

ELECTRÓNICA DIGITAL

La Electrónica estudia el comportamiento de los electrones en diversos medios.

Conocer las leyes que rigen esta forma de comportarse nos permitirá conseguir que éstos hagan lo que nosotros deseemos.

Así, por ejemplo, si construimos un circuito electrónico formado por una pequeña bombilla, una pila y un interruptor y lo conectamos, lograremos que los electrones circulen por todo el circuito y que al atravesar la bombilla parte de ellos se conviertan en luz.

¡Hemos conseguido que los electrones nos obedezcan!

Para dominar a los electrones, es necesario crear circuitos electrónicos, formados por materiales conductores (cables) que unen todos los componentes del circuito, de la misma manera que hay tuberías de agua que recorren nuestras casas, uniendo diferentes elementos: grifos, llaves de paso, el contador del agua...



El objetivo de la electrónica aplicada es construir circuitos electrónicos para que los electrones se comporten de la manera que a nosotros nos interese.

contrar respuestas a preguntas que antes no se podían responder.

Ahora estamos viviendo un momento en el que esa capacidad de manipulación, almacenamiento, recuperación y transporte de la información está creciendo exponencialmente, lo que nos convierte en lo que los sociólogos llaman la "Sociedad de la información", y que tendrá (de hecho, ya tiene) grandes implicaciones sociales.

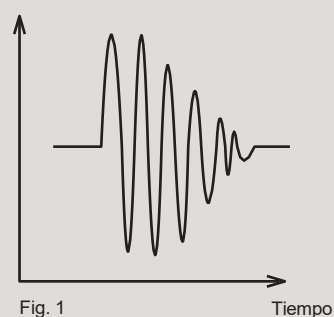
Con la aparición de la electrónica las posibilidades para desarrollar esas capacidades aumentaron considerablemente.

Para comprender los principios de la electrónica analógica, nos centraremos en un ejemplo concreto:

LA MANIPULACIÓN, ALMACENAMIENTO, RECUPERACIÓN Y EL TRANSPORTE DE UNA VOZ HUMANA.

Cuando hablamos, nuestras cuerdas vocales vibran de una determinada manera, lo que origina que las moléculas del aire también lo hagan, chocando unas con otras y propagando esta vibración.

Si no existiesen esas moléculas, como ocurre en el espacio, el sonido no se podría propagar.



TIPOS DE ELECTRÓNICA

Electrónica Analógica

Uno de los grandes retos del hombre es el de manipular, almacenar, recuperar y transportar la información que tenemos del mundo en el que vivimos, lo que nos faculta para ir progresando poco a poco, cada vez con más avances tecnológicos, que facilitan nuestra vida y que nos permiten en-

Si medimos la vibración de una de estas moléculas, durante un espacio corto de tiempo, y la dibujamos, podría tener un aspecto como el que se muestra en la figura 1.

A esta vibración la llamaremos señal acústica.

Cuando ésta señal acústica incide sobre un micrófono, aparece una señal eléctrica que tiene una forma análoga a la de la señal acústica.

Las vibraciones de las moléculas se han convertido en variaciones del voltaje, que al final se traducen en vibraciones de los electrones. Es decir, que con los micrófonos lo que conseguimos es que los electrones vibren de una manera análoga a cómo lo hacen las moléculas del aire (ver figura 1).

Esta nueva señal eléctrica que aparece, se denomina señal analógica, puesto que es análoga a la señal acústica original.

De esta manera, con señales eléctricas conseguimos imitar las señales del mundo real.

Y lo que es más interesante, conseguimos que la información que se encuentra en la vibración de las moléculas del aire, pase a los electrones.

Cuanto mejor sea el micrófono, más se parecerá la señal eléctrica a la acústica, y la información se habrá copiado con más fidelidad.

La electrónica analógica trata con este tipo de señales, análogas a las que hay en el mundo real, modificando sus características (Ejemplo amplificándola, atenuándola, o filtrándola).

Fijémonos en el siguiente esquema:

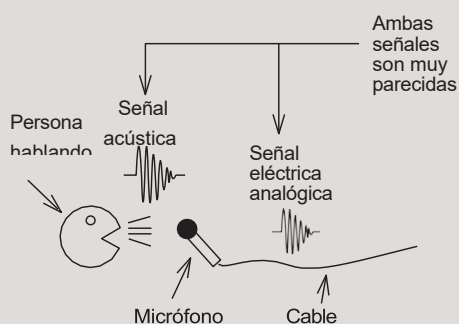


Fig. 2

La persona que habla emite una señal acústica que es convertida en una señal electrónica analógica por el micrófono.

Estas dos señales son muy parecidas, pero la que sale del micrófono es más pequeña.

Por ello se introduce en un circuito electrónico, llamado amplificador, que la agranda.

A continuación, esta señal se puede registrar en una cinta magnética de audio.

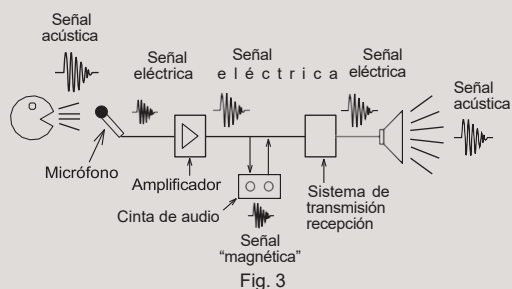


Fig. 3

Lo que se graba es una copia de la señal, pero ahora convertida en señal magnética.

En cualquier momento la señal se puede recuperar, convirtiéndose de señal magnética nuevamente a señal eléctrica.

Una parte del sistema se ha llamado sistema de transmisión-recepción indicándose con esto que la señal eléctrica se puede transportar (por ejemplo, mediante una red telefónica) Fig. 3.

Finalmente se introduce por un altavoz que realiza la conversión inversa: pasar de una señal eléctrica a una acústica que se puede escuchar.

Los problemas de los sistemas analógicos son:

1. La información está ligada a la forma de la onda. Si se degrada, se pierde información.
2. Cada tipo de señal analógica necesita de unos circuitos electrónicos particulares. No es lo mismo un sistema electrónico para audio que para vídeo, puesto que las señales tienen unas características completamente diferentes.

1ª Conclusión:

En las señales analógicas, la información se encuentra en la forma de la onda.

Electrónica digital

Existe otra manera de modificar, almacenar, recuperar y transportar las señales, solucionando los problemas anteriores.



Es un enfoque completamente diferente, que se basa en convertir las señales en números.

Existe un teorema matemático (teorema de muestreo de Nyquist) que nos garantiza que cualquier señal **se puede representar mediante números**, y que con estos números se puede reconstruir la señal original.

De esta manera, una señal digital, es una señal que está descrita por números.

Es un conjunto de números.

Y la **electrónica digital** es la que trabaja con señales digitales, o sea, con números.

Son los números los que se manipulan, almacenan, recuperan y transportan.

Reflexionemos un poco.

Estamos acostumbrados a escuchar el término televisión digital, o radio digital.

¿Qué significa esto?

¡Significa que lo que nos están enviando son números!

Que la información que nos envían está en los propios números y no en la forma que tenga la señal que recibidos.

¿Y qué es un sistema digital?

Un sistema que trabaja con números.

¿Y un circuito digital?

Un circuito electrónico que trabaja con números.

¡Sólo con números!

Si nos fijamos, con un ordenador, que es un sistema digital, podemos escuchar música o ver películas.

La información que está almacenada en el disco duro son números.

En la figura de abajo se muestra un sistema digital.

La **señal acústica** se convierte en una **señal eléctrica**, y a través de un **convertor analógico-digital** se transforma en números, que son procesados por un **circuito digital** y finalmente convertidos de nuevo en una **señal electrónica**, a través de un **convertor digital-analógico**, que al atravesar el altavoz se convierte en una **señal acústica**.

El utilizar circuitos y sistemas que trabajen sólo con números tiene una ventaja muy importante:

Se pueden realizar manipulaciones con independencia de la señal que se esté introduciendo, datos, voz, vídeo...

Un ejemplo muy claro es Internet.

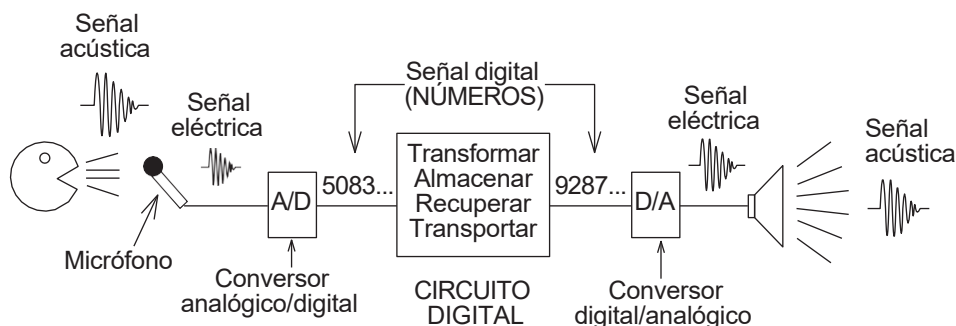
Internet es una red digital, especializada en la transmisión de números.

Y esos números pueden ser datos, canciones, vídeos, programas, etc.

La red no sabe qué tipo de señal transporta, sólo ve números.

2ª Conclusión:

La electrónica digital trabaja con números. La información está en los números y no en la forma de señal. Cualquier señal siempre se puede convertir a números y recuperarse posteriormente.

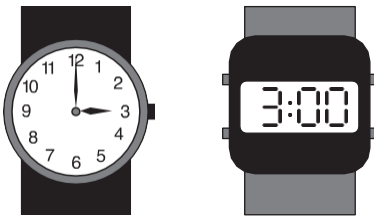


INSISTIMOS EN DESTACAR DIFERENCIAS ENTRE SISTEMAS ANALÓGICOS Y DIGITALES.



Enfoquemos lo expuesto anteriormente de otra forma, buscando una mayor comprensión.

Concepto de analógico y digital (diferencias)



Algunas mediciones pueden representarse en forma **"analógica"** o en forma **"digital"**.

El término **"Digital"** se refiere a cantidades discretas como la cantidad de personas en una sala, cantidad de libros en una biblioteca, cantidad de autos en una zona de estacionamiento, etc.

Los Sistemas digitales tienen una alta importancia en la tecnología moderna, especialmente en la computación y sistemas de control automático.

La tecnología digital se puede manifestar en los siguientes campos:

- Mecánico
- Electromecánico
- Neumático
- Hidráulico
- Electrónico

El término **"Analógico"** se refiere a las magnitudes o valores que varían con el tiempo en forma continua como la distancia y la temperatura, la velocidad, que podrían variar muy lenta o muy rápida como un sistema de audio.

En la vida cotidiana el tiempo se representa en forma analógica por relojes (de agujas), y en forma discreta (digital) por displays digitales.

En la tecnología analógica es muy difícil almacenar, manipular, comparar, calcular y recuperar información con exactitud cuando ésta ha sido guardada, en cambio en la tecnología digital (ordenadores, por ejemplo), se pueden hacer tareas muy rápidamente, muy exactas, muy precisas y sin detenerse.

La electrónica moderna usa para realizar muchas funciones que antes desempeñaba la electrónica analógica.

Un ejemplo muy evidente es el hecho de que la música actualmente se graba en discos compactos (CD's), que previamente se ha convertido en formato digital de su formato original que es analógico.

El equipo creado para reproducir la música grabada de esta manera está lleno de circuitos lógicos digitales.

Sin embargo, los discos de acetato (los discos de 45 r.p.m. y L.P.s. de color negro) utilizaban una aguja que recorría los surcos en el disco para poder reproducir la música grabada en forma analógica.

Nadie duda de la calidad de los discos compactos de hoy, pues tienen un sonido excelente.

La electrónica moderna usa electrónica digital para realizar muchas funciones.

Aunque los circuitos electrónicos pueden resultar muy complejos, en realidad se construyen de un número muy grande de circuitos muy simples.

En un circuito digital se transmite información binaria (ceros y unos) entre estos circuitos y se consigue un circuito complejo con la combinación de bloques de circuitos simples.

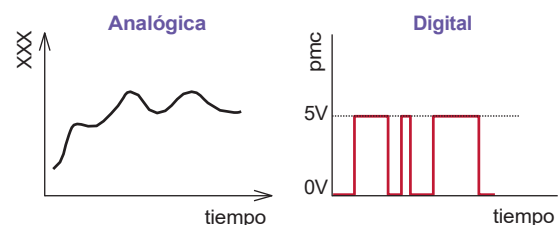


Fig. 6

La información binaria se representa en la forma de "0" y "1", un interruptor "abierto" o "cerrado", "On" y "Off", "falso" o "verdadero", en donde "0" representa falso y "1" verdadero.

Los circuitos lógicos se pueden representar de muchas maneras.

En el circuito siguiente la lámpara puede estar encendida o apagada ("On" o "Off"), dependiendo de la posición del interruptor (Apagado o encendido).

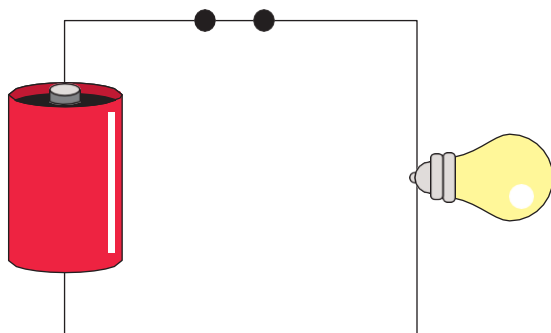


Fig. 7

Los posibles estados del interruptor o interruptores que afectan un circuito se pueden representar en una tabla de verdad.

Las **tablas de verdad** pueden tener muchas columnas, pero todas las tablas funcionan de igual forma.

Hay siempre una columna de salida que representa el resultado de todas las posibles combinaciones de las entradas.

Tabla de verdad:

Columna(s) de entrada; Columna de salida

Entrada (interruptor)	Salida (lámpara)
Abierto	Apagado
Cerrado	Encendido

El número de columnas en una tabla de verdad depende de cuantas entradas hay + 1 (la columna de la salida), el número de filas representa la cantidad de combinaciones en las entradas.

Número de combinaciones = 2^n , donde n es el número de columnas de la tabla de verdad (menos la columna de salida).

Ejemplo: en una tabla hay 3 columnas de entrada lo que producirá: $2^3 = 8$ combinaciones (8 filas).

Un circuito con 3 interruptores de entrada (con estados binarios "0" o "1"), tendrá 8 posibles combinaciones.

Siendo el resultado (la columna salida) determinado por el estado de los interruptores de entrada.

La electrónica digital es uno de los campos de la electrónica aplicada, junto a la analógica, la de potencia, la microelectrónica y la instrumentación.

Es frecuente usar ésta como reclamo publicitario y sinónimo de calidad, sin ser necesariamente cierto, existen sistemas analógicos de gran precisión y fiabilidad que básicamente se diferencian de los digitales, en la señal que procesan.

Características de los sistemas electrónicos digitales:

- a) Circuitos más complejos que los analógicos.
- b) Facilidad para memorizar variables eléctricas.
- c) Se pueden programar, sin cambiar el circuito sólo cambiando la memoria.
- d) Permiten montajes más sencillos y rápidos debido a su alta integración.

Aplicaciones:

- a) Control industrial (autómatas programables).
- b) Proceso o Tratamiento de datos (informática).
- c) Electrodomésticos (electrónica de consumo).

Conclusión final

Una señal es analógica cuando su intensidad o voltaje se representan mediante variables continuas.

En los circuitos y sistemas electrónicos analógicos, se procesan señales analógicas, es decir, señales eléctricas que toman infinitos valores dentro de un intervalo de tiempo como se aprecia en la figura 6.

En los circuitos y sistemas electrónicos digitales se procesan señales eléctricas que toman solo dos valores en el tiempo: 0 y 1,

por lo que la señal cambia por saltos como se muestra.

Los valores cero y uno representan estados lógicos.

Electrónicamente se le asigna a cada uno un voltaje o rango de voltaje determinado:

0 = Nivel bajo de tensión, apagado, Falso, no conduce, no

1 = Nivel alto de tensión, encendido, Verdadero, conduce, sí

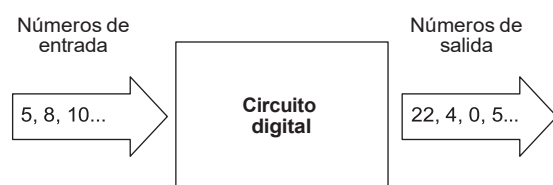


Si deseamos introducirnos a fondo en el mundo de los sistemas digitales, bien estará conocer cómo se “cuenta” en ellos, y con qué unidades.

Circuitos y sistemas digitales

Ya podemos entender de lo que trata esta entrega.

En ella hablaremos de **circuitos digitales**, que manipulan números. Existen unos **números en la entrada** y nuestro circuito generará otros **números de salida**.



Algunos números se considerarán como datos y otros se usarán para el control del propio circuito.

No nos preocuparemos de dónde vienen estos números, pero ya sabemos que o bien vendrán de otro sistema digital, o bien de una señal analógica que se ha convertido a números (se ha digitalizado).

Un circuito digital realiza manipulaciones sobre los números de entrada y genera unos números de salida.

Acabamos de ver cómo un **circuito digital trabaja con números** y sólo con números.

El esquema general de estos circuitos se puede ver en la figura anterior.

Antes de entrar en la comprensión y diseño de estos circuitos, hay que estudiar **cómo se pueden representar esos números**, de manera que el circuito los entienda.

Veremos que existen muchísimas formas de representar el mismo número, pero sólo unas pocas son las que nos interesarán para los circuitos digitales.

Conceptos

El concepto de número todos lo tenemos, pero **un mismo número se puede representar de muchas maneras**.

Por ejemplo, el número 10, lo representamos mediante dos dígitos, el '1' y el '0'.

Si utilizásemos numeración romana, este mismo número lo representaríamos sólo con un único dígito 'X'.

Pero está claro que ambas representaciones, "10" y "X" hacen referencia al mismo número diez.

Estamos acostumbrados a representar los números utilizando diez dígitos: '0', '1', '2', '3', '4', '5', '6', '7', '8', '9'.

Por eso nuestro sistema de representación se denomina **sistema decimal** o **sistema en base diez**.

Analicemos un poco más en detalle el sistema decimal, que es el que manejamos habitualmente.

Vamos a representar el número tres mil doscientos ochenta y uno:

3281

Observamos lo siguiente:

Está constituido por cuatro dígitos: '3', '2', '8' y '1'.

El **orden** en el que están colocados **es muy importante** y si se modifica, se está representando otro número.

Cuanto más a la izquierda está un dígito, más importante es.

Este último aspecto es muy interesante.

Imaginemos que el número 3281 representa el sueldo mensual de un profesional.

Si le preguntamos qué dígito es el que le gustaría modificar para tener un sueldo mayor, no dudaría en señalar al '3'.

"¡Ojalá me subieran en sueldo a 4281 euros!", pensaría.

Sin embargo, se echaría a reír si su jefe le dijese: "te subimos el sueldo a 3285 euros".

El dígito '3' es más importante que todos los que tiene a su derecha. Tiene un **peso mayor** que el resto de dígitos.

De hecho, este dígito '3' está representando al número tres mil.

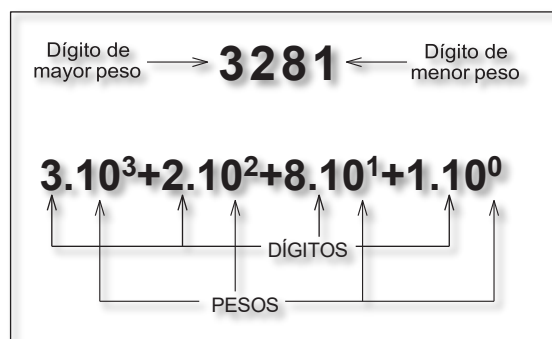
El dígito '2' por estar en tercera posición comenzado desde la derecha, representa el número doscientos, el '8' al ochenta y el '1' al uno.

Podemos descomponer el número de la siguiente manera:

$$\begin{aligned} 3281 &= 3000 + 200 + 80 + 1 = \\ &= 3 \times 1000 + 2 \times 100 + 8 \times 10 + 1 = \\ &= 3 \times 10^3 + 2 \times 10^2 + 8 \times 10^1 + 1 \times 10^0 \end{aligned}$$

Observamos que cada dígito está multiplicando una potencia de 10. Cuanto más a la izquierda se sitúe el dígito, mayor será la potencia de diez por la que se multiplica.

En la figura que sigue se muestra el número 3281 descompuesto en dígitos y pesos, y se indica cuál es el dígito de mayor peso y cuál es el de menor.



Este sistema de representación también se llama **sistema en base diez** porque los pesos de los dígitos son potencias de 10:

El dígito del extremo derecho tiene un peso de 10^0 , los siguientes tienen pesos de 10^1 , 10^2 , 10^3 , 10^4 ...

Normalmente representamos los números en el sistema decimal, que consta de diez dígitos diferentes, asignándoles un peso que es una potencia de diez, y que será mayor cuanto más a la izquierda se encuentre el dígito.

Sistema de numeración binario

El sistema de numeración binario utiliza sólo dos dígitos, el **cero** (0) y el **uno** (1).

Es el que se utiliza en los ordenadores, pues trabajan internamente con dos niveles de voltaje, por lo que su sistema de numeración natural es el sistema binario (encendido 1, apagado 0).

En una cifra binaria, cada dígito tiene distinto valor dependiendo de la posición que ocupe. El valor de cada posición es el de una potencia de **base 2**, elevada a un exponente igual a la posición del dígito menos uno. Se puede observar que, tal y como ocurría con el sistema decimal, la base de la potencia coincide con la cantidad de dígitos utilizados (2) para representar los números.

De acuerdo con estas reglas, el número binario **1011** tiene un valor que se calcula así:

$$\begin{aligned} 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0, \text{ es decir:} \\ 8 + 0 + 2 + 1 = 11 \end{aligned}$$

y para expresar que ambas cifras describen la misma cantidad lo escribimos así:

$$1011_2 = 11_{10}$$

Conversión entre números decimales y binarios

Convertir un número decimal al sistema binario es muy sencillo: basta con realizar **divisiones sucesivas por 2** y escribir los restos obtenidos en cada división **en orden inverso** al que han sido obtenidos.

Por ejemplo, para convertir al sistema binario el número 77_{10} haremos una serie de divisiones que arrojarán los restos siguientes:

$$77: 2 = 38 \text{ Resto: } 1$$

$$38: 2 = 19 \text{ Resto: } 0$$

$$19: 2 = 9 \text{ Resto: } 1$$

$$9: 2 = 4 \text{ Resto: } 1$$

$$4: 2 = 2 \text{ Resto: } 0$$

$$2: 2 = 1 \text{ Resto: } 0$$

$$1: 2 = 0 \text{ Resto: } 1$$

Y, tomando los restos en orden inverso obtenemos la cifra binaria:

$$77_{10} = 1001101_2$$

Volveremos a insistir en el sistema binario.

¿Qué nos impide que utilicemos unos sistemas de representación en los que los pesos de los dígitos, o incluso los dígitos sean diferentes de los del sistema decimal?

Nada.

Por ejemplo, podemos emplear un sistema de representación octal (Base 8), que utiliza sólo ocho dígitos (0, 1, 2...7) para representar cualquier número y los pesos de los diferentes dígitos serán potencias de 8.

En este sistema, si escribimos los dígitos 352 no se corresponden con el número "trescientos cincuenta y dos".

Para calcular cuál es el número que representa hay que multiplicar cada dígito por su correspondiente peso, obteniendo el número equivalente en el sistema decimal.

$$352 = 3 \times 8^2 + 5 \times 8^1 + 2 \times 8^0 =$$

$$3 \times 64 + 5 \times 8 + 2 \times 1 = 234$$

El número 352 en representación octal es equivalente al número 234 del sistema decimal.

En el sistema octal, los dígitos tienen pesos que son potencias de 8, en lugar

de potencias de 10 como en el sistema decimal.

Para evitar confusiones cuando se trabaja con sistemas de representación diferentes, se emplea la siguiente notación:

$$352_8 = 234_{10}$$

El subíndice 8 indica que el número está representado en un sistema octal y con el subíndice 10 se indica que lo está en un sistema decimal.

Resumen de algunos sistemas de representación.

Sistema octal (Base 8).

Utiliza ocho dígitos: 0, 1, 2, 3, 4, 5, 6 y 7 y los pesos son potencias de 8.

Sistema binario (Base 2)

Este sistema de representación sólo utiliza los dígitos 0 y 1 para representar cualquier número.

Fijémonos en lo interesante que resulta esto, ¡sólo con dos dígitos podemos representar cualquiera de los infinitos números!

En el sistema binario los pesos de estos dígitos son potencias de 2.

Veamos un ejemplo del número binario 101001

$$101001 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 2^5 + 2^3 + 2^0 = 41$$

El número binario 101001 se corresponde con el número 41 en decimal.



El sistema binario tiene mucha importancia y se utiliza constantemente en electrónica digital.

Fijémonos en lo que significa esta forma de representación. Utilizando sólo dos dígitos, es posible representar cualquiera de los infinitos números.

En la tecnología actual disponemos de un elemento, llamado **transistor**, que se puede encontrar en dos estados diferentes, abierto (corte) o cerrado (cerrado), a los que le asociamos los dígitos 0 y 1.

Todos los circuitos integrados o chips se basan en estos transistores y trabajan internamente en binario.

Todas las operaciones se realizan utilizando este sistema de representación, por eso es muy importante conocerlo, para entender cómo funcionan los microprocesadores y los chips por dentro.

El peso de los dígitos es una potencia de 2.

Sistema hexadecimal (Base 16)

¿Y sería posible utilizar más de 10 dígitos para representar los números?

También es posible.

Ese es el caso del **sistema hexadecimal**, en el que se emplean **16 dígitos**: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E y F, donde las letras representan los números 10, 11, 12, 13, 14 y 15 respectivamente.

Los pesos de los dígitos son potencias de 16.

Por ejemplo, el número hexadecimal FE2A₁₆ se puede descomponer de la siguiente manera:

$$\begin{aligned} \text{FE2A}_{16} &= F \times 16^3 + E \times 16^2 + 2 \times 16^1 + A \times 16^0 \\ &= 15 \times 16^3 + 14 \times 16^2 + 10 \times 16^0 = 65006_{10} \end{aligned}$$

El **sistema hexadecimal** es muy curioso.

Permite escribir números como los siguientes:

CACA, DE, BACA.

Este sistema, como veremos más adelante, se emplea para escribir números binarios de una manera más compacta, dado que el paso de hexadecimal a binario y viceversa es inmediato.

En este trabajo sólo nos centraremos en el sistema binario.

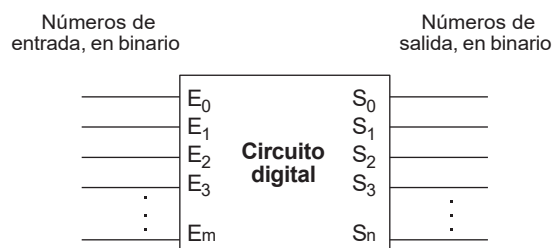
Tabla de conversión para los sistemas decimal - binario - hexadecimal

La tabla que se muestra a continuación representa las equivalencias entre diferentes números expresados en los sistemas **decimal**, **binario** y **hexadecimal**, que son los que más usaremos.

DECIMAL	BINARIO	HEXADECIMAL
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Circuitos digitales y el Sistema binario

Ahora que ya tenemos un poco más claro el concepto de número y las diferentes formas que tenemos de representarlo, podemos retomar el esquema de un circuito digital (ver figura 9) para precisarlo un poco más.



Circuito digital genérico, con entradas y salidas binarias

Con la tecnología que hay actualmente, los **circuitos digitales** manipulan números que están representados en binario. Así podemos decir que un **circuito digital** actual **tiene como entradas y salidas números en binario**.

Es decir, números que vienen expresados con los dígitos '0' y '1'. En la figura anterior se ha dibujado un circuito digital genérico, en el que sus entradas y salidas se expresan en binario. Cada una de las entradas y salida representa un dígito binario.

¿Pero cuál es el peso de este dígito? Eso nos lo indican los subíndices de las letras E y S. Así, la entrada E_0 se corresponde con el dígito de menor peso, la entrada E_1 con los dígitos de peso $2^1 = 2$, y así sucesivamente hasta la entrada n que es la de mayor peso. Lo mismo es aplicable a la salida.

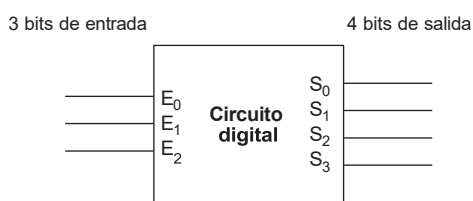
En los circuitos digitales, los números que se procesan, están expresados en binario, tanto en la entrada como en la salida.

Un **dígito binario**, que puede ser '0' ó '1', recibe el nombre de **BIT**, del término inglés **B**inary **digi**T (dígito binario).

Utilizaremos los bits para indicar el tamaño de las entradas y salidas de nuestros circuitos.

Así por ejemplo podemos tener un circuito digital con 3 bits de entrada y 4 de salida.

Este circuito se muestra en la siguiente figura.



Circuito digital con tres bits de entrada y 4 de salida



Los circuitos digitales sólo saben trabajar con números en binario, sin embargo, a los humanos nos es más cómodo trabajar en decimal. Trabajar con número binarios puede parecer poco intuitivo.

Vamos a ver cómo en determinadas ocasiones resulta muy intuitivo el trabajar con números binarios.

Imaginemos que en una habitación hay 5 bombillas situadas en la misma línea, y que cada una de ellas puede estar encendida o apagada.

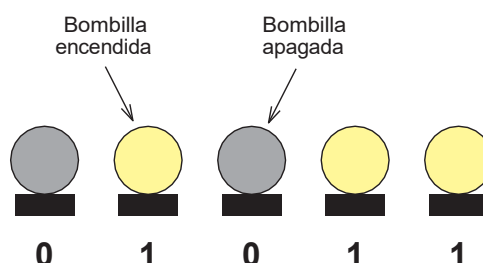
¿Cómo podríamos representar el estado de estas 5 bombillas mediante números?

Una manera muy entendible sería utilizar el **sistema binario**, en el que utilizaríamos el **dígito 1** para indicar que **la bombilla está encendida** y el **dígito 0** para indicar que **está apagada**.

Así el número 01011 nos indica que la primera bombilla está apagada, la segunda encendida, la tercera apagada y las dos últimas encendidas, como se muestra en la figura.

Ésta forma de representar el estado de las bombillas es bastante expresiva.

Es un ejemplo en el que se puede apreciar que pensar en binario resulta más fácil que hacerlo directamente en decimal.



Sistema binario y sistema hexadecimal.

El **sistema hexadecimal** se utiliza para **representar números binarios de una forma más compacta** porque llega un momento en que se hace excesiva la sucesión de ceros y de unos.

Como cada dígito hexadecimal codifica 4 bits, un número hexadecimal de 4 bits permite representar un número binario de 16 bits.

Veamos un ejemplo:

$$1011000111101101_2 = B1ED_{16}$$

Podemos apreciar cómo es mucho más cómodo utilizar el número hexadecimal que el binario.

Pero, ¿cómo se pasa de binario a hexadecimal o viceversa?

El proceso es muy sencillo.

Lo único que hay que conocer es la tabla de conversión para los sistemas decimal-binario-hexadecimal (presentada anteriormente).

El número en binario hay que dividirlo en grupos de 4 bits empezando desde la derecha.

La conversión del número binario anterior se haría de la siguiente manera:

1011	0001	1110	1101
B	1	E	D

Bits y electrónica.

Todavía nos queda una cosa por resolver.

En la electrónica trabajamos con electrones, forzándolos a que hagan lo que nosotros queremos.

En el caso de los circuitos digitales, lo que hacemos es operar con números. ¿Cómo conseguimos esto? ¿Cómo introducimos los números en los circuitos digitales?

La solución es **asignar un voltaje a cada uno de los dos estados de un bit**.

Lo normal, conocido como lógica TTL, es asignar el valor de 5 voltios al dígito '1' y 0 voltios al dígito '0'.

Esta asignación de valores depende de la tecnología empleada.

En la figura 12 se muestra un circuito digital que tiene un bit de entrada.

Si queremos introducir un dígito '1' ponemos la interrupción en la posición A, de manera que por la entrada E llegan 5 voltios.

Si queremos introducir un dígito '0' ponemos el interruptor en la posición B, por lo que llegan cero voltios.

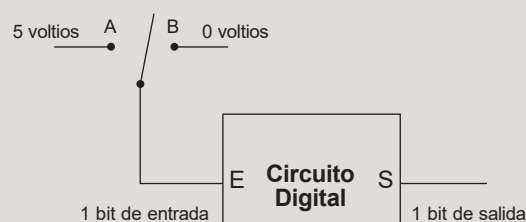


Fig. 12

En los circuitos digitales, se usan dos tensiones diferentes, una para representar el dígito '1' y otra para representar el dígito '0'. Se usan 5 voltios para el dígito '1' y 0 voltios para el dígito '0'.

Otros sistemas de representación.

Para representar los números hemos visto que los circuitos digitales utilizan el sistema binario.

Y hemos estado utilizando el **sistema binario natural**, en el que los bits tienen de peso potencias de 2, que es lo más habitual.

Sin embargo, existen otros sistemas de representación que son binarios en el sentido de que sólo usan los dos dígitos '0' y '1', sin embargo, tienen pesos diferentes.

Algunos de estos sistemas, también conocidos como códigos son los siguientes:

1. Código BCD: Decimal Codificado en Binario.

Es una manera de representar números decimales en binario.

A cada dígito decimal se le asignan 4 bits, correspondientes a su número binario natural.

Así por ejemplo para representar número decimal 21 en BCD, utilizaremos en total 8 bits, 4 para uno de los dos dígitos:

$$21 = 0010\ 0001$$

Los primeros 4 bits representan al dígito '2' y los 4 siguientes al dígito '1'.

2. **Código AIKEN:** Similar al BCD, pero con los pesos cambiados. Cada dígito decimal se representa mediante 4 bits, siendo los pesos de estos bits: 2, 4, 2 y 1.

3. **Código GRAY:** Es una familia de códigos que se caracterizan porque el paso de un número al siguiente implica que sólo se modifica un bit.

Terminología

BIT

Dígito binario. Un bit puede tomar los valores 0 ó 1. Es la abreviatura de las palabras inglesas de Binary digiT.

BYTE

Conjunto de 8 bits. Equivalen en el alfabeto digital a una letra. El número más alto que se puede representar es el 11111111, que en decimal es 255.

KILOBYTE

1.000 BYTES (2 kilobytes son una página completa de texto y 100 kilobytes una foto de teléfono móvil).

MEGABYTE

1.000 KILOBYTES (1Megabyte poco menos que un disquete de 3,5", 10 Megabyte, casi un minuto de vídeo).

GIGABYTE

1.000 Megabytes (1Gigabyte, una hora de vídeo con calidad de imagen y 5 Gigabytes 7.000 fotografías con buena resolución).

TERABYTE

1.000 Gigabytes (7 Terabytes ocupan los archivos de texto que necesitarían 1.000 árboles convertidos en papel).

PETABYTE

1.000 Terabytes.

EXABYTE

1.000 Petabytes.

ZETTABYTE

1.000 Exabytes.

YOTTABYTE

1.000 Zettabytes.

EJERCICIOS RESUELTOS

1. **Descomponer el número 63 en sus dígitos y pesos.**

Solución:

$$63 = 6 \times 10^1 + 3 \times 10^0$$

Dígitos: '6' y '3' con pesos 10 y 1.

2. **Hacer lo mismo que en ejercicio 1, pero con el número 10358.**

Solución:

$$10358 = 1 \times 10^4 + 0 \times 10^3 + 3 \times 10^2 + 5 \times 10^1 + 8 \times 10^0$$

Dígitos '1', '0', '3', '5' y '8' con pesos 10000, 1000, 100, 10 y 1 respectivamente.

3. **Pasar los siguientes números al sistema decimal:**

$$a) 1010111_2$$

$$\begin{aligned} \text{Solución: } 1010111 &= 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= 64 + 16 + 4 + 2 + 1 = 87 \end{aligned}$$

$$b) BABA_{16}$$

$$\begin{aligned} \text{Solución: } BABA &= B \times 16^3 + A \times 16^2 + B \times 16^1 + A \times 16^0 \\ &= 11 \times 4096 + 10 \times 256 + 11 \times 16 + 10 \\ &= 45056 + 2560 + 176 + 10 = 47802 \end{aligned}$$

c) 29_8

Solución: $29_8 = 2 \times 8^1 + 9 \times 8^0 = 2 \times 8 + 9 = 25$

4. Pasar de hexadecimal a binario:

a) FFFF

Solución:

FFFF = F - F - F - F = 1111-1111-
1111-1111 = 1111111111111111

b) FA00

Solución:

FA00 = F-A-0-0 = 1111-1010-0000-
0000 = 1111101000000000

c) 321C

Solución:

321C = 3-2-1-C = 0011-0010-0001-
1100 = 11001000011100

LÓGICA BINARIA. OPERACIONES LÓGICAS CON VARIABLES BINARIAS

Función IGUAL

Función NO

Función OR

Función AND

Función NOR

Función NAND

Función XOR

Función XNOR

Lógica binaria. Operaciones lógicas.

Una operación lógica es una consideración que asigna un valor (cierto o falso) a la combinación de condiciones (cierto o falso) de uno o más factores.

Los factores que intervienen en una operación lógica sólo pueden ser ciertos o falsos.

Y por ello, el resultado de una operación lógica puede ser únicamente, cierto o falso.

Por ejemplo, imaginemos el sistema de control del toldo de una cafetería, que se gobierna mediante una operación lógica.

Sabemos que existe un enemigo acérrimo del toldo que es el viento y para controlarlo el sistema incorpora un pequeño anemómetro tal como el de la figura.



Anemómetro tal como se instala



Anemómetro con interruptor crepuscular



Toldo con anemómetro e interruptor crepuscular separado

Al anemómetro suele acompañarle un interruptor crepuscular.

El anemómetro gira si hace aire y lleva solidario a su eje un generador de tensión o dinamo.

La tensión que genera activa cualquier sistema ya estudiado, como un relé, contactor o un transistor en corte o paso, interrumpiendo la continuidad del circuito de activación del toldo.

Para que el motor que extiende el toldo se accione deberá tener en cuenta tres factores: ¿Es de día? ¿Hace sol? ¿Hay viento?

O sea, que las premisas para que el toldo se extienda son:

- Que sea de día.
- Que luzca el sol.
- Y que no haya viento.

Si los tres factores son ciertos, el motor debe ponerse en marcha y extender el toldo.

De Día	Sol	Viento	Toldo
Falso	Falso	Cierto	Falso
Falso	Falso	Cierto	Falso
Cierto	Falso	Falso	Falso
Cierto	Cierto	Falso	Cierto

Los resultados de una operación lógica, para cada uno de los valores posibles de las variables, se fijan en una tabla denominada **Tabla de Verdad**, como la del ejemplo anterior.

Para que un procesador pueda ejecutar las operaciones lógicas, es preciso asignar un valor binario a cada una de las condiciones posibles.

Hemos leído que se suele asignar un (1) al valor **cierto** y un (0) al valor **falso**, con el criterio denominado lógica positiva.



Las operaciones lógicas más importantes son: **IGUAL** (idéntico), **NO** (negación), **OR** (O), **AND** (Y), **NOR** (O negada), **NAND** (Y negada), **XOR** (O exclusiva) y **XNOR** (O exclusiva negada).

Veámoslas con detalle:

Función IGUAL.

El resultado S de aplicar la función lógica **igual**, sobre una variable **a**, es muy simple: si **a** es **cierto** (1), es cierto S y si **a** es **falso** (0), S también es **falso** (0).

Estos dos resultados posibles se muestran en la tabla de verdad adjunta:

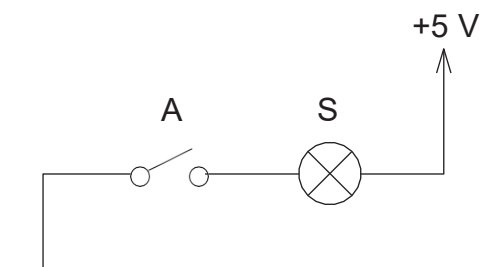
a	S
1	1
0	0

Un ejemplo sencillo, de aplicación práctica de esta función lógica, sería el encendido de las luces del alumbrado público.

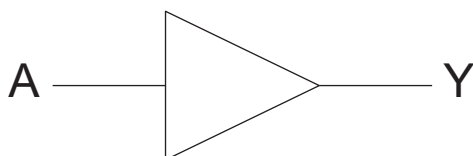
En algún lugar de la ciudad se instala un detector crepuscular, que denuncia cuándo es de noche y activa un interruptor que enciende las farolas de las calles.

Si es de noche (1) se encienden las lámparas (1); si NO es de noche (0) NO se encienden las lámparas (0).

Un circuito eléctrico capaz de realizar esta función lógica es el siguiente:



Se puede representar así:



Función NO

El resultado **S** de aplicar la función lógica **NO**, sobre una variable **a**, es muy simple: si **a es cierto (1)**, **S es falso (0)** y si **a es falso (0)**, **S es cierto (1)**.

Estos dos resultados posibles se muestran en la tabla de verdad adjunta.

Se conoce también como función negación: **S** equivale a una **a** negada.

a	S
1	0
0	1

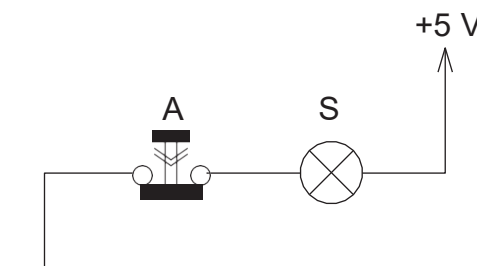
Podemos expresarlo así: $S = a$

También puede expresarse así: $S = a'$

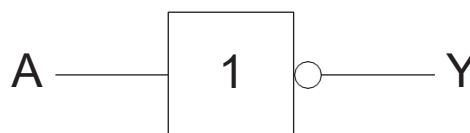
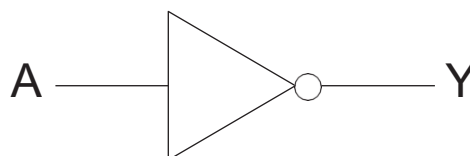
Un ejemplo sencillo de aplicación práctica de esta función lógica sería el circuito que controla el acceso a una oficina bancaria, a través de una puerta automática equipada con un detector de metales que activa un interruptor.

Si el detector de metales **SI** nota que el cliente lleva objetos metálicos (1), la puerta **NO** se abre (0).

En cambio, si el cliente **NO** lleva objetos metálicos (0), la puerta **SI** se abre (1).



Podemos representarla así:



Función OR

La función **OR** equivale a la conjunción disyuntiva **O**.

El resultado **S** de aplicar la función lógica **OR**, sobre dos variables **a** y **b** es el siguiente: **S** es cierto si **a es cierto (1)** o si **b es cierto (1)**.

Cuando se aplica una operación lógica sobre 2 variables caben 4 combinaciones posibles.

Los resultados de la operación lógica **OR**, en las cuatro combinaciones posibles de valores dos variables, se muestran en la tabla de verdad adjunta.

S es cierto si **a** es **cierto** (1) o si **b** es **cierto** (1)

a	b	$S = a + b$
0	0	0
0	1	1
1	0	1
1	1	1

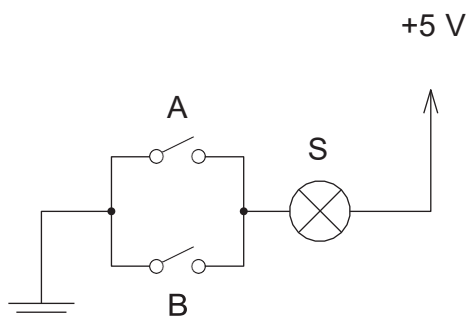
Una aplicación práctica sencilla de la operación lógica **OR**, sería el circuito de señalización instalado en un comercio, en el que se puede entrar por dos puertas distintas, que avisaría al dependiente al entrar un cliente por cualquiera de las dos puertas del establecimiento.

Si un cliente entra por la puerta **a** (1) o si un cliente entra por la puerta **b** (1), el timbre **SI** suena (1).

Si no entra ningún cliente por ninguna de las puertas **a** (0) ni **b** (0), el timbre **NO** suena (0).

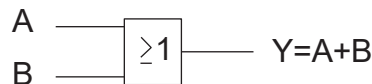
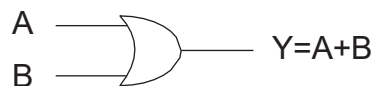
Un circuito eléctrico compuesto por dos interruptores en paralelo, cumple la lógica **OR**. La lámpara **S** se encenderá (1) si se acciona el interruptor **a** (1), o si se acciona el interruptor **b** (1), o si se accionan ambos interruptores.

Si no se acciona ningún interruptor, la lámpara **NO** se encenderá (0).



Una aplicación de la operación lógica **AND**, sería el sistema de control de los pasajeros en un aeropuerto.

Otra forma de representarla:



Conviene indicar que, además de las puertas de dos entradas, también se fabrican puertas OR de tres y cuatro entradas, en las que la salida tomará el nivel lógico 0 (nivel bajo de tensión) únicamente cuando todas sus entradas estén también a nivel lógico 0.

Función AND

La función AND equivale a la conjunción copulativa Y: El resultado, S de aplicar la función lógica AND, sobre dos variables **a** y **b** es el siguiente:

S es **cierto** si **a** es **cierto** (1) y si **b** es **cierto** (1).

Los resultados de la operación lógica AND, en las cuatro combinaciones posibles de valores dos variables, se muestran en la tabla de verdad adjunta.

a	b	$S = a \cdot b$
0	0	0
0	1	0
1	0	0
1	1	1

Cada pasajero debe pasar por dos controles:

¿Tiene tarjeta de embarque?

¿No lleva objetos metálicos peligrosos?

Una empleada del aeropuerto comprueba que tiene un billete válido y le da una tarjeta de embarque.

A continuación, un grupo de agentes comprueban su equipaje de mano con un escáner y un arco detector de metales.

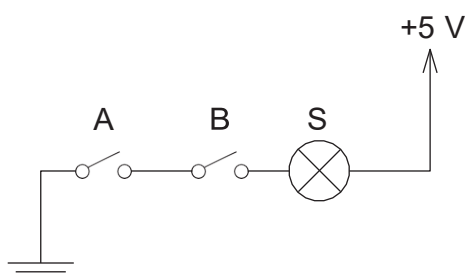
El pasajero sólo puede embarcar en el avión si tiene tarjeta de embarque (1) y no lleva consigo objetos peligrosos (1).

En los demás casos no puede embarcar.

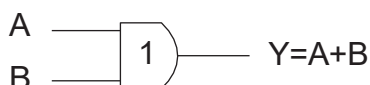
Es fácil construir un circuito eléctrico que cumple la lógica **AND**: dos interruptores en serie, **a** y **b**, por ejemplo.

La lámpara **S** sólo se encenderá si se actúa sobre el interruptor **a** (1) y sobre el interruptor **b** (1).

En todos los demás casos, la lámpara **NO** se encenderá.



Podemos representarla así:



Al igual que ocurría con la puerta OR, con- viene indicar que, además de las puertas de dos entradas, también se fabrican puertas AND de tres y cuatro entra- das, en las que la salida tomará el nivel lógico 1 (nivel alto de tensión) únicamente cuando to- das sus entradas estén también a nivel lógico 1.

Función NOR

La función **NOR** equivale a la función **OR** negada.

El resultado **S** de aplicar la función lógica **NOR**, sobre dos variables **a** y **b** es el siguiente:

S es **cierto** si **a** es **falso** (0) y si **b** es **falso** (0).

Los resultados de la operación lógica **NOR**, en las cuatro combinaciones posibles de valores dos variables, se muestran en la tabla de verdad adjunta:

a	b	$S = \overline{a + b}$
0	0	0
0	1	0
1	0	0
1	1	1

Una aplicación práctica sencilla de la operación lógica **NOR**, sería el sistema de seguridad de un puente levadizo.

Un detector **a** se activa cuando circula un vehículo en el puente por el carril derecho.

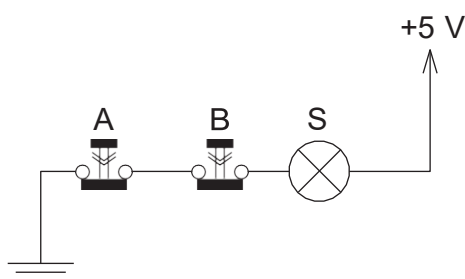
Otro detector **b** se activa cuando circula otro vehículo por el carril contrario.

Los motores que accionan el sistema de elevación del puente sólo deben ponerse en marcha si se da la condición **NOR**: no hay ningún vehículo circulando por el carril derecho NI por el carril izquierdo.

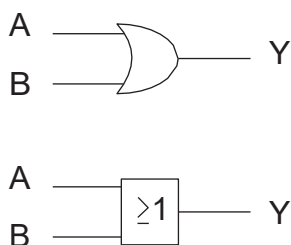
Un circuito eléctrico compuesto por dos interruptores normalmente cerrados, en serie, cumple la lógica **NOR**:

La lámpara **SI** se encenderá (1) si **NO** se acciona el interruptor **a** (0) **NI** se acciona el interruptor **b** (0).

Si se acciona cualquiera de los dos interruptores, la lámpara **NO** se encenderá (0).



Podemos ponerla así:



Conviene indicar que cuando a una puerta NOR o NAND se le unen todas sus entradas, éstas se comportan como una puerta inversora, por lo que con estos dos tipos de puertas lógicas es posible realizar cualquier circuito lógico.

Función NAND

La función **NAND** equivale a la función **AND** **negada**.

El resultado **S** de aplicar la función lógica **NAND**, sobre dos variables **a** y **b** es el siguiente:

S es **cierto** si **a** es **falso** (0) o si **b** es **falso** (0) o si son **falsas** ambas variables.

Los resultados de la operación lógica **NAND**, en las cuatro combinaciones posibles de valores dos variables, se muestran en la tabla de verdad adjunta:

a	b	$S = \overline{a \times b}$
0	0	1
0	1	1
1	0	1
1	1	0

Una aplicación práctica sencilla de la operación lógica **NAND** sería, por ejemplo, el control del aire acondicionado de un edificio inteligente.

Supongamos que el sistema está equipado de un interruptor horario, mejor astronómico, que dispone de un contacto conmutador.

Durante el día el contacto que utilizamos de este reloj está activado (1) y durante la noche está desactivado (0).

Supongamos también que en cada planta del edificio hay un detector de presencia con una salida, también conmutada, que pone el contacto que utilizamos, a **UNO** (1), cuando todo el mundo ha salido y a **CERO** si hay alguien.

Todos los contactos de los detectores de presencia están en serie.

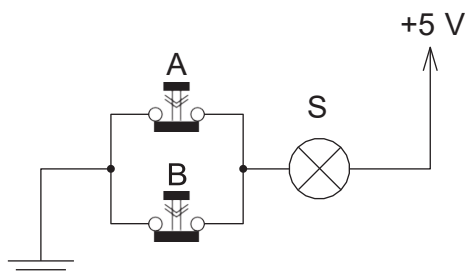
¿Cómo controlar la puesta en marcha del aire acondicionado?

Muy fácil, con un circuito que siga la lógica **NAND**:

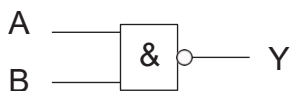
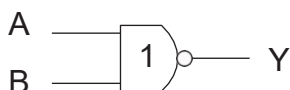
El aire acondicionado se parará cuando sea de noche y no quede nadie en el edificio.

Y se activará cuando sea de día. Aunque no haya gente.

Un circuito eléctrico compuesto por dos interruptores normalmente cerrados, en paralelo, cumple la lógica **NAND**: la lámpara **SI** se encenderá (1) si **NO** se acciona el interruptor **a** (0) o si **NO** se acciona el interruptor **b** (0) o si **NO** se accionan ambos interruptores.



Podemos representar esta puerta como:



Conviene indicar que cuando a una puerta NOR o NAND se le unen todas sus entradas, éstas se comportan como una puerta inversora, por lo que con estos dos tipos de puertas lógicas es posible realizar cualquier circuito lógico.

Otra indicación:

En la actualidad se fabrican puertas NAND hasta de ocho entradas.

Función XOR

La función **XOR** se conoce también con el nombre de **OR EXCLUSIVA**.

El resultado **S** de aplicar la función lógica **XOR**, sobre dos variables **a** y **b** es el siguiente:

S es **cierto** solo si **a** es **cierto** (1) o si **b** es **cierto** (1), pero no si ambas variables son ciertas.

Esto se representa así:

$$S = (A \oplus B)$$

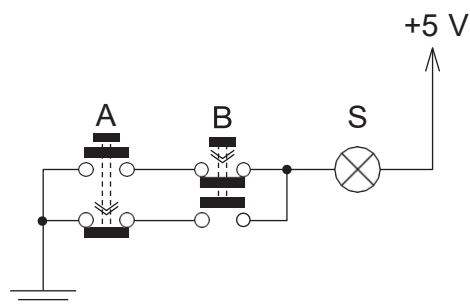
Los resultados de la operación lógica **XOR**, en las cuatro combinaciones posibles de valores dos variables, se muestran en la tabla de verdad adjunta:

a	b	$S = (A \oplus B)$
0	0	0
0	1	1
1	0	1
1	1	0

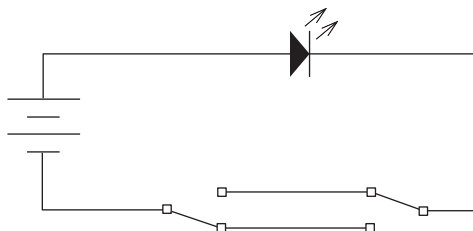
Un circuito eléctrico como el del esquema siguiente, compuesto por dos pulsadores dobles **NA + NC**, cumple la lógica **XOR**:

La lámpara **S** se encenderá (1) **EXCLUSIVAMENTE** si se acciona el pulsador **a** o si se acciona el pulsador **b**, pero **NO** se encenderá si se accionan simultáneamente ambos pulsadores.

Tampoco se encenderá si no se acciona ninguno de los dos pulsadores.



Este circuito se puede asimilar a este otro:



Y un ejemplo lo tenemos en dos conmutadores de la luz de un pasillo:

Se enciende o se apaga la luz accionando uno u otro pulsador, pero nunca se encenderá si se accionan los dos a la vez.

Se puede representar así:



Función XNOR

La función **XNOR** se conoce también con el nombre de **OR EXCLUSIVA NEGADA**.

El resultado **S** de aplicar la función lógica **XNOR**, sobre dos variables **a** y **b** es el siguiente:

S es **CIERTO** si **a** y **b** son ciertos o si **a** y **b** son falsos.

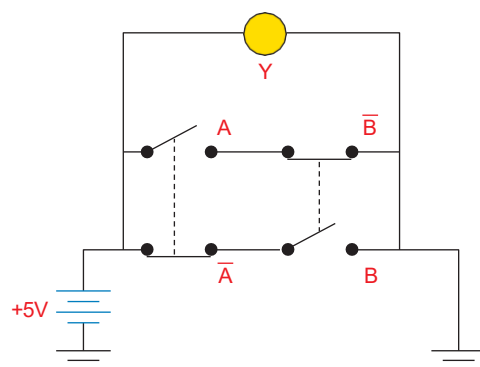
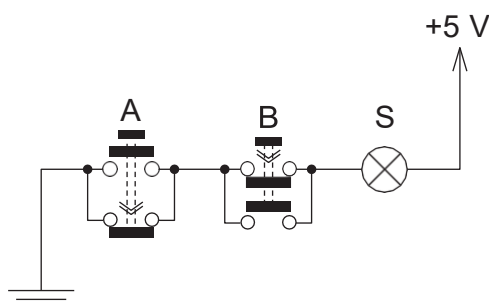
Es decir, si ambas variables tienen el mismo valor.

Y se representa así: $S = (\overline{A \oplus B})$

Los resultados de la operación lógica **XNOR**, en las cuatro combinaciones posibles de valores dos variables, se muestran en la tabla de verdad adjunta:

a	b	$S = (\overline{A \oplus B})$
0	0	1
0	1	0
1	0	0
1	1	1

Un circuito eléctrico como el del esquema siguiente, compuesto por dos pulsadores dobles **NA + NC**, cumple la lógica **XNOR**: la lámpara **S** se encenderá si se accionan **ambos pulsadores** o si **no** se acciona ninguno de ellos.



La puerta sólo da un **uno lógico** a la salida cuando las dos entradas son iguales.

Se puede representar así:

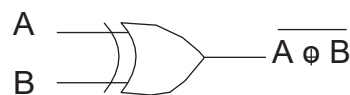

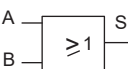
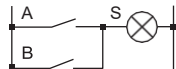
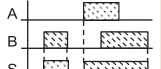


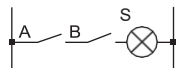
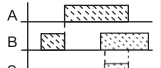
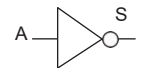

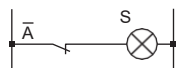
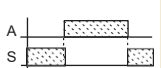

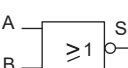
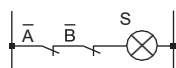
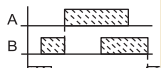

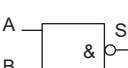
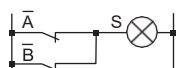
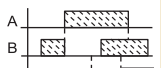

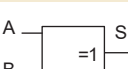
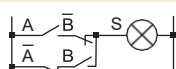
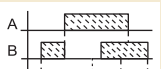


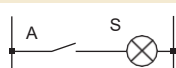



Fig. 32

SIMBOLOS DE LAS PUERTAS

Para tenerlos agrupados y a mano aparecen a continuación la mayor parte de las puertas lógicas.

Función	Ecuación lógica	Símbolos			Tabla de verdad	Cronograma															
		Norma MIL	Norma IEC	Circuito físico con contactos																	
OR	$S = A+B$				<table><tr><th>A</th><th>B</th><th>S</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	S	0	0	0	0	1	1	1	0	1	1	1	1	
A	B	S																			
0	0	0																			
0	1	1																			
1	0	1																			
1	1	1																			
AND	$S = A \cdot B$				<table><tr><th>A</th><th>B</th><th>S</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	S	0	0	0	0	1	0	1	0	0	1	1	1	
A	B	S																			
0	0	0																			
0	1	0																			
1	0	0																			
1	1	1																			
NOT	$S = \bar{A}$	 inversor			<table><tr><th>A</th><th>S</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	S	0	1	1	0										
A	S																				
0	1																				
1	0																				
NOT (OR+NOT)	$S = \overline{A+B}$ $S = \bar{A} \cdot \bar{B}$				<table><tr><th>A</th><th>B</th><th>S</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	S	0	0	1	0	1	0	1	0	0	1	1	0	
A	B	S																			
0	0	1																			
0	1	0																			
1	0	0																			
1	1	0																			
NOT (AND+NOT)	$S = \overline{A \cdot B}$ $S = \bar{A} + \bar{B}$				<table><tr><th>A</th><th>B</th><th>S</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	S	0	0	1	0	1	1	1	0	1	1	1	0	
A	B	S																			
0	0	1																			
0	1	1																			
1	0	1																			
1	1	0																			
EXOR	$S = A \oplus B$ $S = \bar{A}B + A\bar{B}$				<table><tr><th>A</th><th>B</th><th>S</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	S	0	0	0	0	1	1	1	0	1	1	1	0	
A	B	S																			
0	0	0																			
0	1	1																			
1	0	1																			
1	1	0																			
EXCI TADOR	$S = A$				<table><tr><th>A</th><th>S</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	A	S	0	0	1	1										
A	S																				
0	0																				
1	1																				

Observaciones:

Los símbolos se representan de distinta forma según sea la NORMA que rija en el país donde se realice el trabajo.

La **NORMA MIL-STD** es la que utiliza el ejército americano en sus equipos e instalaciones.

La **NORMA IEC** pertenece a Commission Electrotechnique Internationale.

La función lógica estará compuesta por diversas variables lógicas relacionadas entre sí. Dichas operaciones son la suma lógica (+), el producto lógico (*) y la negación (así, a negada la representaremos por \bar{a}).

Sin más preámbulos, veamos cómo se obtiene el circuito digital para que resuelva una función lógica, y qué mejor forma de verlo que con un ejemplo concreto:

Idéese un circuito digital tal que realice la función lógica $G = (a \times b) + [c \times (a + b)]$

Empecemos por ver cuántas variables forman a la función G.

En este caso se ve que son tres, a , b y c .

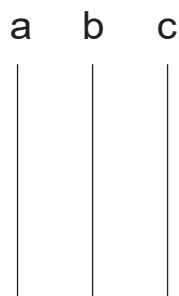
Ya podemos empezar a dibujar el circuito.

Hay que trazar tantas líneas verticales como variables tenga la función, poniéndole a cada una de ellas como título el nombre de una variable:

EJECUCIÓN DE FUNCIONES CON PUERTAS LÓGICAS

Una de las características de la electrónica digital que más gusta al aficionado es lo fácil que resulta iniciarse en el diseño de circuitos.

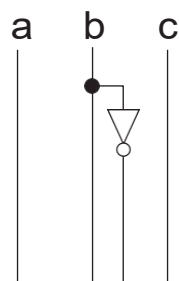
A continuación, vamos a ver lo sencillo que puede llegar a ser diseñar un circuito digital con tal de que conozcamos la función lógica que debe de verificar.



¿Hay alguna variable aislada que esté negada?

Si la respuesta es sí (y en este caso lo es, fíjese en la función, en ella aparece \bar{b}) habrá que colocar una puerta inversora de tal forma que su entrada esté conectada a la línea de la variable que debe negarse.

A la salida de esta puerta tendremos la variable negada:

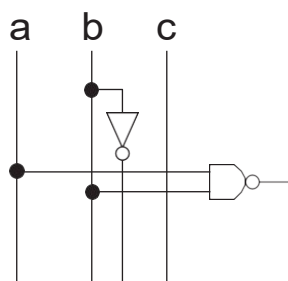


Como puede apreciarse, la salida de la puerta se ha ampliado con una línea vertical más.

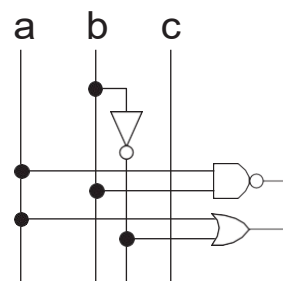
El siguiente y último paso es ir realizando con puertas lógicas las operaciones de la función lógica.

Así, podríamos hacer ahora el producto negado de la variable a con la variable b .

Para ello emplearemos la puerta NAND:

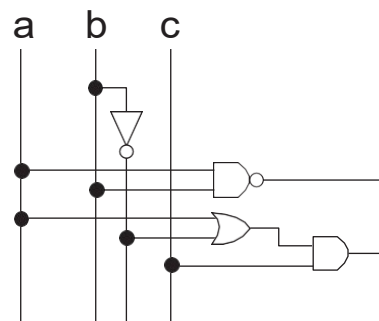


Podríamos seguir con la suma lógica de a con b (puerta OR):



La puerta OR recién colocada entrega a su salida $a + b$.

Si multiplicamos esto por c tendríamos $[c \times (a + b)]$ (ver la expresión de la función G):



Por último sólo queda sumar $(a + b)$ (que está en la salida de la puerta NAND) con $[c \times (a + b)]$ (presente en la salida de la puerta AND) para obtener la función G de salida:

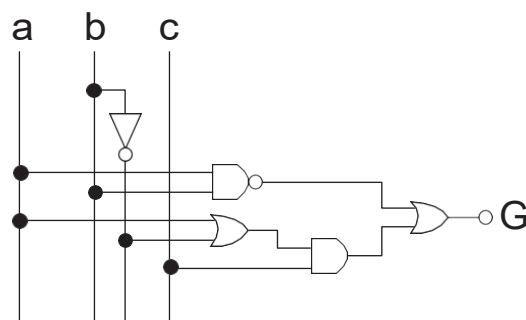


Fig. 39

Y ya tenemos nuestro circuito terminado.

Este circuito calcula automáticamente el valor de la función G para cualquier combinación de valores de las variables que forman la función.



Hemos podido percatarnos de lo siguiente:

Para poder llevar a cabo la ejecución de la función planteada es imprescindible conocer con detalle cada una de las puertas lógicas que existen.

Por este motivo, invitamos a nuestros lectores a repasar la parte que trata de las puertas lógicas.

ANECDOTARIO

Un profesor enseñó a un alumno, para distraer el aburrimiento durante un viaje, cómo se traducía un problema a un circuito eléctrico de control sencillo, compuesto de interruptores.

Y le puso el siguiente ejemplo:

Un padre tenía una hija guapísima (Laura, por ejemplo), a la que rondaban tres pretendientes.

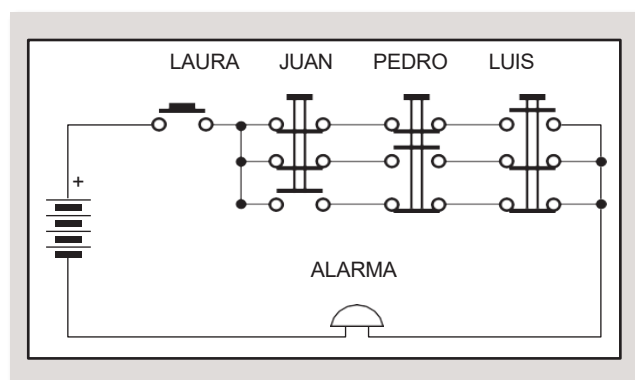
El hombre quería vivir tranquilo y diseñó un sistema, con alarma, que le advirtiese cuando la chica se quedaba a solas con uno sólo de sus pretendientes (pongamos que eran Juan, Pedro y Luis, por ejemplo).

El sistema no debería sonar si la chica estaba con más de uno de ellos porque, creía que, en esas condiciones no podía existir el menor problema.

Traducir esa complicación a una tabla de verdad fue sencillo.

LAURA	JUAN	PEDRO	LUIS	ALARMA
1	0	0	1	1
1	0	1	0	1
1	1	0	0	1

Suponiendo que cada uno de los muchachos se sentaba siempre en la misma butaca, podríamos dotarlas de un grupo de interruptores.



MÁS CONCEPTOS DE INTERÉS

Nos va a ayudar para hacer montajes, conocer la lógica combinatoria y la lógica secuencial.

Lógica combinatoria

Los circuitos combinacionales son aquellos que no tienen en cuenta la variable tiempo.

Están formados por combinaciones de puertas lógicas.

Lógica secuencial

Método de ordenamiento de acciones, razonamiento, y expresión de la automatización de maquinaria, equipos y procesos.

Y su interrelación con el hombre.

Esto nos da por consiguiente los binomios, hombre-máquina, hombre-proceso.

En el proceso productivo se traduce en mecanización de movimiento, y como es lógico suponer existen sistemas híbridos, mecánica-electricidad-electrónica-hidráulica etc. para la obtención de un fin determinado.

Así damos funcionamiento a un método de producción.

En una operación de ensamble existe

también la lógica binaria y su aplicación de decisión, aun cuando se coloquen seis tornillos existe la lógica al colocarlos y en el apriete de los mismos, y el método no debe cambiar.

En una operación de desbaste en maquinado manual o CN la lógica secuencial está presente, desde la colocación de la pieza, como del ataque de la herramienta y velocidad de corte.

Un proceso de tratamiento térmico es un diseño con lógica secuencial, desde la colocación de las piezas dentro del horno, el encendido de los quemadores, la generación de atmósfera si se requiere, el precalentamiento, la zona o tiempo de tratamiento, el proceso de enfriamiento, etc.

El trabajador implicado en el proceso productivo y una vez capacitado en Lógica Secuencial sobre las diferentes máquinas y o equipos productivos, podrá determinar con precisión las causas de una interrupción de su equipo, la detección de un defecto en una

unidad en el proceso, o bien sugerir mejoras en el sistema de control visual del equipo. También es posible que el operador precise qué componente periférico está comenzando a provocar un fallo y anticiparse a un paro.

Báscula, flip-flop o biestable

Cualquiera de los adjetivos es válido. Se trata de un componente "de tiempo" que básicamente consta de dos conexiones:

Clock o Reloj (C). - Se le suele llamar Reloj (Clock en inglés), y es la entrada de control principal.

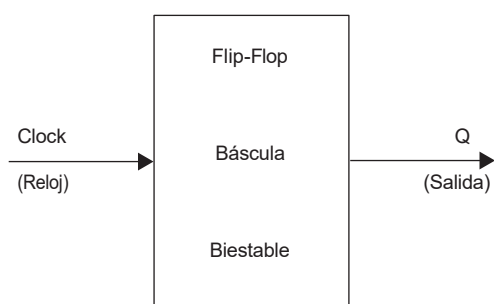


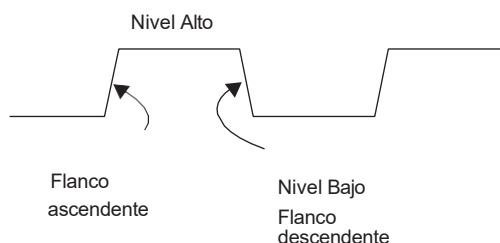
Fig. 41

Salida (Q). - Normalmente se rotula con la letra Q.

Es frecuente que también figure la misma salida, negada, lo que evita tener que añadir puertas inversoras cuando se añaden funciones combinatorias.

Todos los circuitos que pueden tomar diversos estados a lo largo del tiempo, como es éste, llevan, además, al menos una línea cuya activación supone su reposición a un estado cierto (normalmente la salida pasa a valer cero) llamada Reset.

En la modalidad más elemental, la salida del flip-flop (Q) conmuta (si estaba alta se hace baja y viceversa) cada vez que a su entrada (Clock) se le aplica un flanco activo. Ha de quedar clara la idea de flanco, y para ello puede ser de ayuda la figura.



A esta báscula también se le llama **Maestro-Esclavo**. La razón de su nombre es su constitución interna.

El contador

La aplicación más inmediata de la báscula es el contador.

Podemos construir un contador empleando sólo básculas, sin añadir nada más.

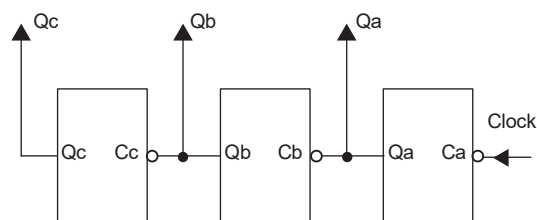
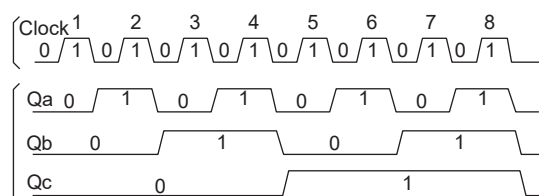


Fig. 43

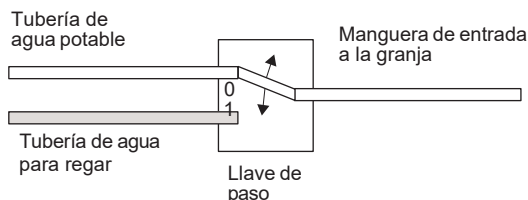
Abajo tenemos el diagrama de estados, y en él salta a la vista cómo la interpretación de los niveles que ofrecen las salidas, debidamente ordenados (bit MSB el de la báscula más alejada a la entrada) es un número binario

que lleva en todo momento la cuenta de la cantidad de pulsos que han ido apareciendo en la entrada principal del circuito (Clock).



Este primer detalle es la razón de que a este circuito se le llame contador.

Multiplexores



Conceptos

Un **Multiplexor** es un **circuito combinacional** al que entran varios canales de datos, **y sólo uno de ellos**, el que hayamos seleccionado, es el que **aparece por la salida**. Es decir, que es un circuito que nos permite **SELECCIONAR** qué datos pasan a través de dicho componente.

Vamos a ver un ejemplo NO electrónico.

Imaginemos que hay dos tuberías (canales de datos) por el que circulan distintos fluidos (datos).

Una transporta agua para regar y la otra agua potable.

Estas tuberías llegan a una granja, en la cual hay una única manguera por la que va a salir el agua (bien potable o bien para regar), según lo que seleccione el granjero posicionando la **llave de paso** en una u otra posición.

En la figura se muestra un esquema.

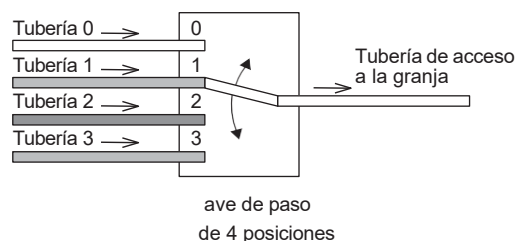
Las posiciones son la 0 para el agua potable y 1 para el agua de regar.

Moviendo la llave de paso, el granjero puede seleccionar si lo que quiere que salga por la manguera es agua potable, para dar de beber al ganado, o agua para regar los cultivos.

Según cómo se posicione esta llave de paso, en la posición 0 ó en la 1, seleccionamos una tubería u otra.

Pero ¿por qué sólo dos tuberías? Porque es un ejemplo. A la granja podrían llegar 4 tuberías.

En este caso el granjero tendría una llave de paso con 4 posiciones, como se muestra en la figura 46.



Esta llave se podría poner en 4 posiciones distintas para dar paso a la tubería 0, 1, 2 ó 3. Obsérvese que sólo pasa una de las tuberías en cada momento, ¡y sólo una!

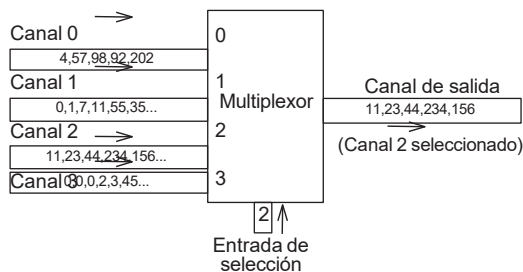
Hasta que el granjero no vuelva a cambiar la llave de paso no se seleccionará otra tubería.

Con este ejemplo es muy fácil entender la idea de **multiplexor**.

Es como una **llave de paso**, que sólo conecta uno de los canales de datos de entrada con el canal de datos de salida.

Ahora en vez de en tuberías, podemos pensar en canales de datos, y tener un esquema como el que se muestra en la figura 47, en la que hay 4 canales de datos, y sólo uno de ellos es seleccionado por el multiplexor para llegar a la salida.

En general, en un multiplexor tenemos dos tipos de entradas:



Entradas de datos: (Las tuberías en el ejemplo).

Entrada de selección: Indica cuál de las entradas se ha seleccionado (posición de la llave de paso).

Entradas de datos

Multiplexores y bits

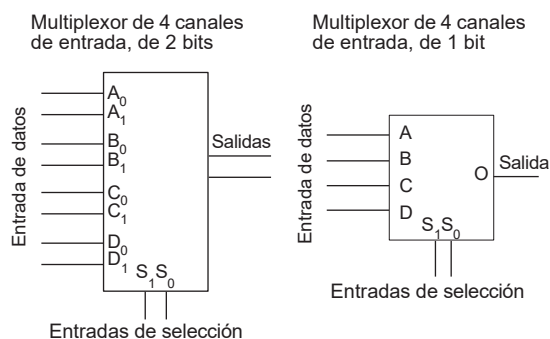
Hemos visto cómo a un **multiplexor** le llegan **números** por distintas entradas y según el número que le llegue por la **entrada de selección**, lo manda por la salida o no.

¡Números!

Recordemos que los circuitos digitales sólo trabajan con números.

Pero estos números, vimos que siempre vendrán expresados en binario y por tanto se podrán expresar mediante **bits**. ¿Cuántos bits? Depende de lo grande que sean los números con los que se quiere trabajar.

En el interior de los microprocesadores es muy normal encontrar multiplexores de 8 bits, que tienen varias entradas de datos de 8 bits. Pero se puede trabajar con multiplexores que tengan 4 bits por cada entrada, o incluso 2, o incluso 1 bit.



En la figura se muestran dos multiplexores que tienen 4 entradas de datos.

Por ello la entrada de selección tiene dos bits (para poder seleccionar entre los cuatro canales posibles).

Sin embargo, en uno las entradas de datos son de 2 bits y en el otro de 1 bit.

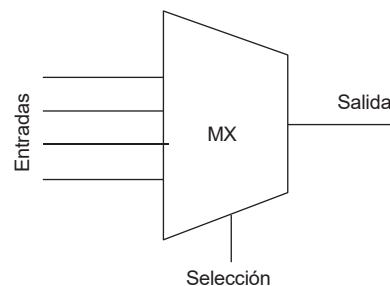
Salidas de datos

Mirando el número de salidas, podemos conocer el tamaño de los canales de entrada.

Así en los dos multiplexores de la figura, vemos que el de la izquierda tiene 2 bits de salida, por tanto sus canales de entrada son de 2 bits. El de la derecha tiene 1 bit de salida, por tanto los canales de 1 bit.

Los multiplexores en lo que principalmente nos centraremos son los que tienen canales de 1 bit. A partir de ellos podremos construir multiplexores mayores, bien con un mayor número de canales de entrada o bien con un mayor número de bits por cada canal.

Representación de un multiplexor. En la figura podemos verificar cómo se representa.



Representación de un multiplexor

Los registros de desplazamiento

Son circuitos secuenciales sencillos, compuestos por un conjunto de biestables que se conectan en serie, y una circuitería adicional que controla los modos de cargar y aceder a los datos que almacenan.

Su principal función, en el seno de sistemas más complejos, es servir de almacenamiento temporal de un conjunto de bits sobre los que se está realizando una tarea de procesamiento.

Las dos operaciones básicas que realiza un registro de desplazamiento es la carga/almacenamiento de los datos y el desplazamiento de estos a lo largo de los biestables que lo componen; la obtención de los datos correctos a la salida del dispositivo depende de una cuidadosa sincronización de las señales de control, de entrada y de salida.

La mejor manera de entender conceptos nuevos es apoyándonos en analogías con temas que nos son familiares.

En este caso no vamos a hacer una excepción, por lo que utilizaremos como ejemplo el funcionamiento de una cola, una fila, como la de un banco o supermercado.

Supongamos que dos tipos de personas pueden formar parte de una cola.

Estos dos tipos de personas son las que se ven en la figura siguiente, y es imposible confundir una con otra.

Es decir, siempre estaremos seguros que en una posición determinada de la fila está una u otra persona.

Las llamaremos "0" (al más grueso) y "1" (al más flaco).

La cola que usaremos como ejemplo tiene 8 lugares, que hemos numerado del 0 al 7, pero nada impide trabajar con colas más largas, por lo que todo lo que se vea aquí, puede ser generalizado para colas de la longitud que se desee.

La cola utilizada como ejemplo tiene 8 posiciones.

Otra particularidad de nuestra hipotética cola es que nunca puede estar vacía.

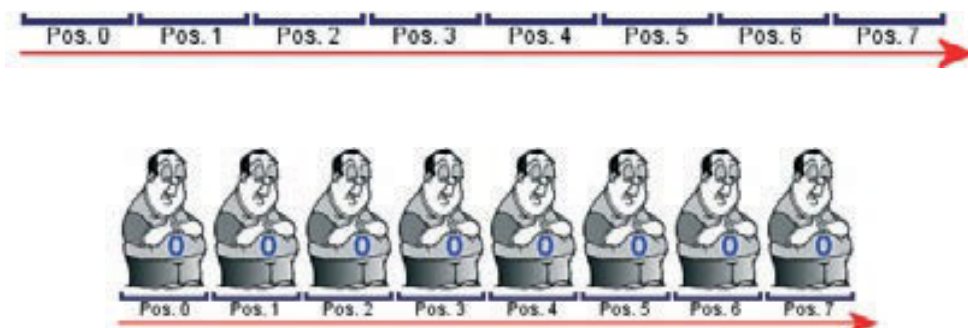
Todas sus posiciones tienen que estar ocupadas, ya sea por "los más gruesos" o "los más flacos".

En el estado inicial, la cola se encuentra completamente llena de "gruesos", como se a continuación en la fig.51

Nuestra cola funciona como cualquier cola de la vida real: cuando alguien nuevo llega a la fila, se coloca en el último lugar, que en este caso corresponde a la "posición 0". Como la cola tiene una longitud máxima (en nuestro ejemplo) de 8 posiciones, para hacer lugar al recién llegado, es necesario que todos los que estaban en la fila "avancen" una posición.



"0" y "1", nuestros personajes.



El que estaba en la posición 0 pasa la 1, el que estaba en la 1 pasa a la 2, y así hasta llegar al que estaba en la posición 7, que "sale" por el extremo opuesto.

Llega un nuevo integrante a la cola, y...
...ocupa el último lugar, desplazando a todos los demás una posición. El primero "sale" de la fila.

Este es el estado final de nuestra fila, con el nuevo integrante en el último lugar.

Si continuaran incorporándose personas en la fila, el proceso se repetiría con cada nuevo integrante que llegue.

Como el que entra primero es el primero en salir, a este tipo de colas se las llama **"FIFO"**, por **First Input, First Output** (Primero que entra, primero que sale).

Con todas estas cuestiones en mente podemos seguir avanzando en la compren-

sión del funcionamiento de los registros de desplazamiento.

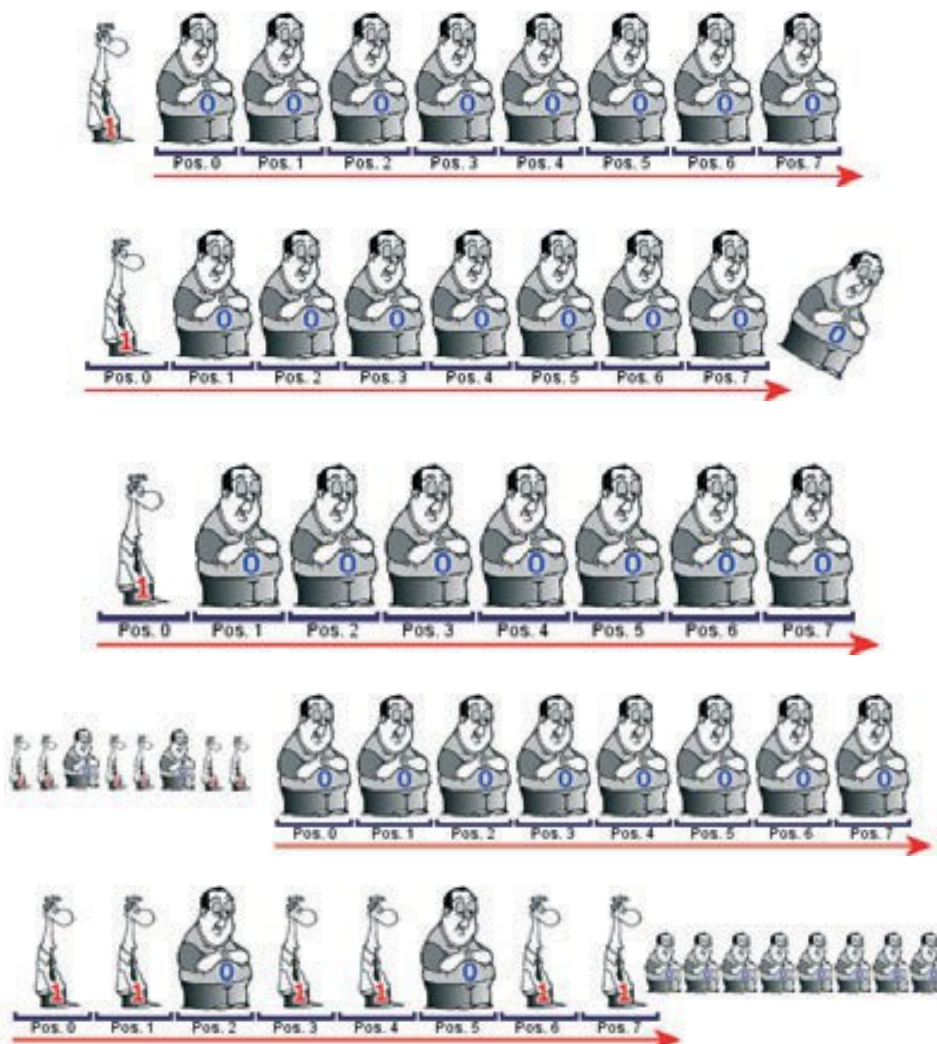
Supongamos que queremos que en la cola haya dos flacos en los primeros lugares, luego un gordo, otra vez dos flacos, luego otro gordo por último dos flacos más (como siempre, 8 personas en total).

Sabiendo que cada personaje que ingresa en la cola "empuja" a toda una posición a la derecha, si queremos que el que termine ocupando el extremo derecho de la cola sea un flaco, ese será el que primero debe entrar.

Siguiendo el razonamiento anterior, los personajes deberían entrar en la fila en el orden siguiente:

Los nuevos integrantes de la fila, esperando para ocupar su lugar.

Este es el estado final de nuestra fila, con los integrantes originales desplazados hacia la derecha.



Poniendo fin a nuestra analogía, tendríamos que los integrantes de esta hipotética cola son los ceros (0) y los unos (1) (o estados altos y bajos) de nuestros circuitos, es decir, nuestros datos.

La cola en si es el registro de desplazamiento.

Cuando decíamos que el estado inicial de la cola eran 8 gordos, estábamos queriendo decir que al alimentar nuestro circuito, todas las salidas estarán en "0" o estado bajo.

Hay una salvedad, y es la existencia del "reloj".

Efectivamente, en un circuito real, los datos pasan al registro de desplazamiento con cada pulso de reloj.

Podemos pensar en este reloj como si se tratase de un "maestro de ceremonias", que da una palmada cada vez que alguien debe ingresar en la cola.

Muchos circuitos de registros de desplazamiento "reales" también incluyen un

sistema de RESET, que permite poner simultáneamente todas las salidas en "0" o estado bajo, sin necesidad de ingresar 8 ceros seguidos.

Esto permite limpiar rápidamente el registro de desplazamiento.

Cuando decimos "*rápidamente*" nos referimos a que como la velocidad de los pulsos del reloj (CLOCK) no puede ser infinita (típicamente el máximo ronda los 10 o 20 MHz) y cada dato demora el tiempo de un pulso de reloj en desplazarse por el registro, introducir 8 "0"s llevaría 800 ns (100 ns x 8 bits), contra los 100 ns que demora en aplicarse el RESET.

No obstante, para obtener los tiempos exactos implicados se debe consultar la hoja de datos del integrado que estemos utilizando, ya que los límites varían incluso con la tensión de alimentación y la temperatura.

CIRCUITOS INTEGRADOS BASADOS EN LOGICA DIGITAL

Un poco de historia para entrar en materia.

Hemos recorrido un largo camino desde los días de los tubos de vacío.

Desde que se construyó el primer microprocesador, en 1971, ha sido posible construir computadores más y más pequeños, permitiendo una gran variedad de inventos.

Pero, ¿Cómo pueden construirse los circuitos integrados en un espacio tan pequeño?

Cuando se conectó el primer computador electrónico del mundo (ENIAC) en la Universidad de Pennsylvania, Filadelfia, en Febrero de 1946, el equipo ocupaba toda una sala

y requería más de 18.000 tubos de vacío para funcionar.

Los tubos de vacío fueron vitales, ya que éstos actuaron como interruptores electrónicos para realizar los cálculos.

El grupo que construyó el ENIAC (Electronic Numerical Integrator and Calculator) se desanimó al comprobar que los tubos se quemaban ya que generaban temperaturas de hasta 200°C.

Y debían ser reemplazados con mucha frecuencia.

¿Qué podían hacer ante esta situación?

Nada, sólo esperar.

Porque, como hemos leído...

Un par de años más tarde, en 1948, William Shockley, John Bardeen y Walter Brattain en los mundialmente famosos Laboratorios Bell en New Jersey inventaron el transistor. A diferencia de los tubos de vacío que eran frágiles y costosos, los transistores resultaron fáciles de fabricar, pequeños, robustos y sobre todo baratos.

En unos pocos años los transistores revolucionaron la forma de construir equipos electrónicos.

Los equipos de radio, de televisión, de navegación, audífonos y equipos médicos fueron más baratos, más fáciles de construir y más avanzados como consecuencia de este invento.

Pero el mayor impacto del transistor estaba aún por llegar.

Al principio, los investigadores hacían transistores de uno en uno, soldándolos en tarjetas de circuito junto con los otros componentes requeridos.

Pero mientras un transistor era altamente resistente, los delgados cables que los conectaban a la tarjeta del circuito eran frágiles y propensos a dañarse.

Así, en 1950, Robert Noyce en Fairchild Semiconductor, una compañía establecida en los alrededores de San José, California (área que sería llamada más tarde Silicón Valley), y Jack Kilby en Texas Instruments en Dallas diseñaron y construyeron el primer circuito en el cual todos los componentes fueron integrados en diferentes capas en un trozo de silicio.

Estos circuitos integrados fueron baratos, fáciles de construir y más

auspiciados y sufragados por el programa espacial Apolo.

En 1970 estos circuitos contenían más de 30.000 componentes en una sola placa de silicio, el "chip".

En 1971, la compañía americana de chips, Intel, dio otro gran paso.

Los ingenieros de Intel pusieron todos los componentes de un computador - la unidad central de procesos, la memoria, los registros de almacenamiento de datos, las unidades de control de entrada y salida- en un minúsculo chip para crear el primer computador en un chip o microprocesador.

El microprocesador de Intel fue realmente notable comparado con el ENIAC, su precio 30.000 veces más barato, consumía una milésima de potencia y podía ser colocado en la punta de un dedo.

Era 200 veces más rápido y tenía alojados 2300 transistores.

Desde entonces, el número de transistores que podía ser comprimido en un chip fue duplicándose cada 18 meses, fenómeno que es conocido como la Ley de Moore, llamada después Gordon Moore, uno de los fundadores de Intel.

El nivel de conocimiento que hoy existe sobre los chips permite almacenar 30 millones o más transistores y es un indicador de que la ley de Moore puede desbaratarse próximamente.



La Ley de Moore expresa que, aproximadamente, cada 18 meses se duplica el número de transistores en un circuito integrado.

Se trata de una ley empírica, formulada por el cofundador de Intel, **Gordon**

E. Moore el 19 de abril de 1965, cuyo cumplimiento se ha podido constatar hasta hoy.

Los microprocesadores han tenido un profundo impacto en la sociedad humana.

Con un microprocesador se puede utilizar el mismo circuito para diferentes aplicaciones.

Lo único que se tiene que hacer es cambiar el programa que manejará este microprocesador.

La construcción compacta de computadores ha hecho posible un amplio rango de dispositivos, tales como los computadores personales, calculadoras de bolsillo, relojes digitales y juegos de vídeo.

Los microchips son tan baratos que muchas fábricas los usan para el control de todo equipo electrónico desde autos y aeronaves hasta máquinas lavadoras y tostadoras.

La idea clave que prevalece en la fabricación de los circuitos integrados es que componentes electrónicos tales como transistores y condensadores se pueden fabricar conectando capas de materiales con diferentes propiedades electrónicas.

Los ingenieros comprendieron que en vez de fabricar los componentes individualmente y unirlos para construir un circuito, el circuito entero podía ser pensado como un conjunto de capas colocadas una encima de otra, razón por la cual usaron la tecnología "complementary metal oxide semiconductor" o "tecnología CMOS para construir los chips".

La base del circuito integrado es silicio ultra puro, una dura sustancia cristalina con una estructura atómica similar al carbón del diamante.

Debido a que no existen partes móviles o componentes delicados el transistor es extremadamente robusto y seguro.

Esto es importante.

Los modernos chips contienen millones de transistores y si un pequeño porcentaje no trabaja, el chip puede quedar inutilizado.

Los transistores no necesitan mucha potencia y generan poco calor, de manera que pueden ser empaquetados en un área muy pequeña.

El tamaño del transistor está determinado por la distancia entre la fuente y el drenaje, que es conocida como tamaño típico.

Hoy día es posible construir transistores con tamaño típico tan pequeño como 0,18 micrómetros, un ciento de veces más delgado que un cabello humano, y serán más pequeños en el futuro.

Los circuitos integrados se beneficiaron enormemente con la reducción del tamaño de los componentes que contienen.

La velocidad a la cual un transistor puede conmutar, o cambiar del estado de conducción al estado de corte, está determinada por la distancia en que fluye la corriente de un lado al otro.

Mientras más pequeños son los transistores más rápidamente trabajan.

El empaquetamiento de más transistores en un área más pequeña significa que los componentes alámbricos son más cortos y ello aumenta la velocidad de operación del chip.

El primer microprocesador realizó 60.000 instrucciones por segundo.

Los más modernos pueden ejecutar un billón.

¿Cómo se construyen los circuitos integrados?

El proceso de construcción comienza con un único gran cristal de silicio, con forma de salchicha, de alrededor de 20 centímetros de espesor, cortada en rodajas.

Se hacen diferentes chips de cada rodaja y se separan al final del proceso de fabricación.

Primero cada rodaja es pulida para dar un acabado casi perfecto y luego es cocida en una atmósfera rica en oxígeno de manera que crece una delgada capa de óxido de silicio, como un vidrio aislante, en la superficie.

Cada capa se hace cubriendo la rodaja con una delgada película de material sensible a la luz, conocido como fotoreistor.

Un fotoreistor tiene la especial propiedad de que la luz cambia su estructura química, permitiéndole reaccionar con otro producto químico y puede ser depuesto lavándolo.

La luz es irradiada sobre el fotoreistor a través de una plantilla o máscara según sea la forma requerida.

Esto deja al material inferior en el patrón requerido, listo para el siguiente paso.

Este proceso es conocido como litografía modelada (patterned lithography).

Reducir el tamaño de las figuras que se pueden definir por el método litográfico es uno de los mayores desafíos que enfrentan los fabricantes de chips, ya que mientras los transistores sean empaquetados más densamente resultan más rápidos y los chips son más complicados.

El factor que limita el tamaño de la figura es la longitud de onda de la luz proyectada a través de la plantilla al fotoreistor.

Mientras más pequeña es la longitud de onda más pequeña es la figura que se puede definir.

Los fabricantes de chips han logrado usar luz ultravioleta para definir figuras de hasta 130 nanómetros de espesor.

Pero para reducir aún más la longitud de onda es necesario desplazarse al

espectro de los rayos X, que son difíciles de producir y manejar.

Además, estos rayos atraviesan los materiales sin afectarlos, haciendo la fabricación de plantillas más difícil.

Alternativas a los rayos X incluyen la litografía de emisión de electrones y de iones, en la que la longitud de onda asociada con electrones y con iones se usa para definir las figuras.

Después que el fotoreistor expuesto a la luz se ha arrancado, el material inferior revelado puede ser tratado de una de las tres formas siguientes: por dopaje, deposición o corrosión.

Dopaje es el proceso en el cual los átomos de fósforo o boro se adhieren a la estructura del silicio creando ya sea un semiconductor tipo P o un semiconductor tipo N.

A estos átomos se les quita electrones para crear iones y luego acelerarlos hacia la superficie del chip a alta velocidad.

Estos chips se calientan y se procede a enfriarlos lentamente para evitar daños causados por su bombardeo y permitir a los recién llegados difundirse en la estructura.

Esta difusión puede ser un problema mayor para los fabricantes si ocurre posteriormente, cuando el chip está siendo usado en un computador, es decir, si los átomos se “secan” demasiado rápido se pueden conectar con otras regiones de chips, cambiando las propiedades del semiconductor y causando un mal funcionamiento.

Con el fotoreistor ubicado en su lugar, los átomos dopantes se adhieren a la parte irradiada del chip.

El fotoreistor puede ser entonces destituido completamente a través del lavado.

Deposición es el proceso de crear una delgada película en el chip.

Una de las maneras más comunes de hacer esto es el denominado chisporroteo (sputtering), un proceso en el cual el material a ser depositado es bombardeado por un chorro de iones que causa que los átomos y moléculas floten en su superficie.

A estos átomos y moléculas se les permite depositarse sobre el chip.

Con el patrón fotoresistor en su lugar constituyen una delgada capa con la forma requerida.

El chisporroteo es usado para poner capas de materiales aisladores tales como el dióxido de silicio, o capas de materiales conductores de metal como el aluminio o el cobre.

Corrosión es una manera de eliminar selectivamente materiales de la superficie para descubrir lo que está debajo.

La corrosión es usualmente tratada por exposición de la rodaja a un gas altamente ionizado conocido como plasma.

El plasma reacciona con la superficie y físicamente saca los átomos como un arenador dejando la capa de abajo expuesta en la forma deseada.

Cuando se completa uno de estos procesos, el resto del fotoresistor es removido y se prepara la próxima capa.

Es común para los modernos chips requerir hasta 30 capas y hasta 600 pasos de fabricación.

Aunque esto suena como exagerado, es poco comparado a los millones de componentes que se ha logrado montar en un chip.

Finalmente, la rodaja es cubierta en un revestimiento protector de dióxido de silicio y nitruro de silicio, antes de que se pruebe cada chip para asegurarse que trabaja bien.

La rodaja es entonces cortada en chips separados.

Los defectuosos son apartados y

los chips buenos son colocados en encapsulados plásticos.

El encapsulado es una parte importante del proceso y debe proveer las uniones de conectores externos a alambres extremadamente finos que conectan los contactos de entrada y de salida de varios chips.

El encapsulado permite al chip ser conectado fácilmente en una tarjeta de circuitos conocida como tarjeta madre (motherboard), y es la que conecta al chip con el resto del computador y provee protección de los contaminantes externos y golpes del uso diario.

La complejidad de los chips significa que diseñarlos es una tarea titánica.

Desde que se inició el desarrollo de los chips, el proceso ha ido cambiando desde aquel que se realizaba enteramente en forma manual hasta el que se hace ahora sustancialmente automático.

En efecto, los circuitos son ahora tan complejos que es muy difícil hacerlos manualmente.

Se ha desarrollado un poderoso software industrial que provee un sistema de diseño por computador para la fabricación de chips.

El proceso de diseño está por sí mismo dividido en un número de niveles que permite al diseñador ver a los chips de diferentes maneras.

Por ejemplo, un diseñador puede especificar el propósito de las diferentes partes del circuito.

Este nivel es conocido como diseño funcional.

El diseño lógico determina los pasos matemáticos necesarios para ejecutar el diseño funcional y los componentes necesarios para realizarlos.

¿Cuál es la velocidad de un chip?

Las diferentes partes de un chip deben ser coordinadas de tal forma que

las operaciones lógicas se realicen en el orden correcto.

Cada chip posee un reloj interno que genera una señal regular alternando un voltaje alto y bajo.

La velocidad del reloj actúa como un cronómetro, sincronizando las acciones del resto del chip como si fuera un director de orquesta.

La velocidad es usualmente medida en MHz, que significa un millón de veces por segundo.

Así un chip de 500 MHz tiene un reloj que genera una señal con quinientos millones de voltajes altos y bajos en un segundo.

En general, la mayor velocidad de un tipo particular de chip, produce una mayor cantidad de cálculos.

Por ejemplo, un chip Pentium III de 750 MHz será más rápido que un chip Pentium III de 500.

Hoy día los transistores operan con billones de electrones que pasan a través de ellos.

La conducta de estos electrones es como la de las bolas de billar y se puede predecir y simular fácilmente.

Cuando un gran número de electrones pasa a través de un transistor puede pensarse como un estado "on" o estado "1", en lenguaje binario.

Cuando unos pocos electrones pasan por un transistor puede pensarse como un estado "off" o estado "0".

Los investigadores están intentando construir computadores cuánticos y predicen que serán útiles para criptografía, supercálculos y para simular el Universo con más detalle que antes.



¿De qué sirve todo lo que hemos comentado sobre puertas lógicas?

Sirve para construir circuitos.

Pero...

¿Cómo los realizamos físicamente?

Las **puertas lógicas** se encuentran encapsuladas dentro de **circuitos integrados** o también conocidos como **chips**.

En la figura se muestra una foto de dos de ellos, junto a una moneda de 1 euro para apreciar su tamaño.

Coloquialmente, entre alumnos, reciben el nombre de "cucarachas", porque son negros y tienen patas.

Hay una familia de circuitos integrados, **74XX**, que está estandarizada de manera que se ha definido la información que entra o sale por cada una de las patas.

Así pueden existir multitud de fabricantes, pero todos respetando el mismo estándar. En la figura siguiente (Fig. 58) se muestra un esquema del integrado 7402, que contiene en su interior 4 puertas NOR de dos entradas.

Por las patas denominadas VCC y GND se introduce la alimentación del chip, que normalmente será de 5V, aunque esto depende de la tecnología empleada.

Por el resto de patas entra o sale información binaria codificada según la tecnología empleada.

Por ejemplo se puede asociar 5V al dígito '1' y 0V al dígito '0'.

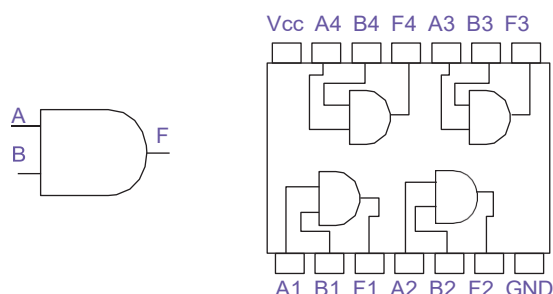
A la hora de **realizar un diseño**, estos chips se insertan en una placa y se interconectan las patas con el resto de chips o partes de nuestro circuito.

La interconexión se realiza por medio de cables.

Cuando se realiza una **placa profesional**, las interconexiones entre los chips son **pistas de cobre** en la superficie de la placa.

Estas placas reciben el nombre de **placas**

de circuito impreso, o por sus siglas en inglés **PCB** (printed circuit Board).



CARACTERÍSTICAS	
Técnica Digital	Técnica Analógica
<ul style="list-style-type: none"> • Sólo son posibles tensiones “High” y “Low” • Gran escala de integración • Alta seguridad • Ausencia de interferencias 	<ul style="list-style-type: none"> • Es posible cualquier valor de tensión • Problemas de ajuste y distorsión • Influencia de señales por interferencia

Las tecnologías utilizadas son:

- TTL: Lógica – transistor – transistor
- CMOS
- ECL: Lógica Emisores acoplados

Etapas de desarrollo de los circuitos integrados

Nivel de integración	Número de funciones por chip	Ejemplo de aplicaciones
1965: S SI (Small Scale Integration)	>100	Circuitos básicos compuerta AND, compuerta OR, compuerta NAND, compuerta NOT, compuerta NOR, Otras
1968: MSI (Medium Scale Integration)	de 100 a 1000	Registros, contadores
1972: LSI (Large Scale Integration)	de 1000 a 10000	Microprocesadores, memorias
1976: VLSI (Very Large Scale Integration)	de 10000 a 100000	Microprocesadores completos
1980: VVLSI (Very Very Large Scale Integration)	> 100000	Microprocesadores múltiples incluyendo memoria, puertos de entrada y salida

En el diseño de los circuitos digitales fundamentalmente se emplean 3 “herramientas”, o puertas lógicas, aunque luego veremos que hay más:

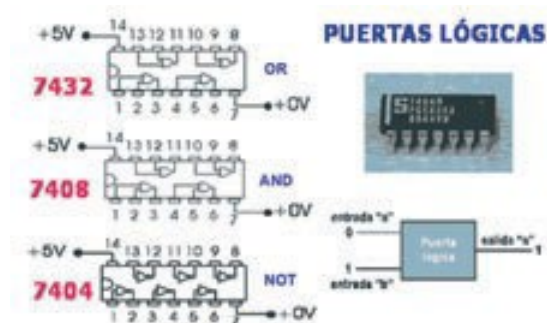


Fig. 59

TECNOLOGIA TTL (Transistor-Transistor Logic)

Compuertas Lógicas

Las compuertas lógicas son bloques de construcción básica de los sistemas digitales; operan con números binarios, por lo que se les denomina puertas lógicas binarias.

En los circuitos digitales todos los voltajes, a excepción de las fuentes de alimentación, se agrupan en dos posibles categorías: voltajes altos y voltajes bajos.

Todos los sistemas digitales se construyen utilizando básicamente tres compuertas lógicas básicas, estas son las AND, OR, y NOT ; o la combinación de estas.

¿Qué es TTL?

Acrónimo inglés de Transistor-Transistor Logic o “Lógica Transistor a Transistor”. Tecnología de construcción de circuitos electrónicos digitales, en los que los elementos de entrada de la red

lógica son transistores, así como los elementos de salida del dispositivo.

Características de los TTL

La familia de circuitos integrados TTL tiene las siguientes características:

- La tensión o voltaje de alimentación es de + 5 Voltios, con $V_{\min} = 4,75$ Voltios y $V_{\max} = 5,25$ Voltios.
- Su realización (fabricación) es con transistores bipolares multiemisor.
- La velocidad de transmisión entre los estados lógicos es su mejor ventaja, ya que esta característica le hace aumentar su consumo.
- Su compuerta básica es la NAND

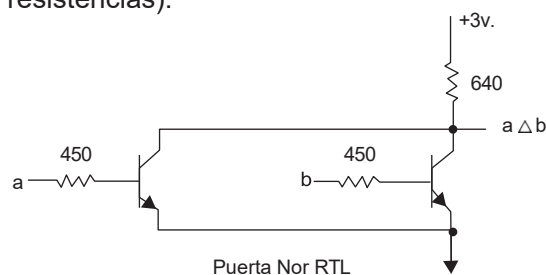
Familia de los Circuitos Lógicos Integrados

	Tecnología	Serie
Familia de circuitos lógicos integrados con transistores bipolares	TTL	TTL estándar TTL de baja potencia TTL Shootky TTL Shootky de baja potencia TTL avanzada
	ECL	

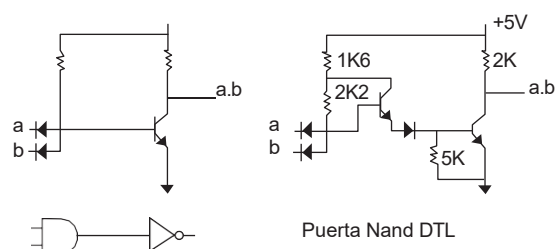
	Tecnología	Serie
Familia de circuitos lógicos integrados con transistores MOSFET	CMOS	CMOS estándar CMOS HC CMOS HCT
	NMOS	
	PMOS	
BiCMOS Combina transistores bipolares con transistores MOSFET		

Antecedentes. Primeras familias lógicas: CI con transistores bipolares

Las primeras puertas lógicas integradas eran mera copia directa de las puertas "**O-NEGADA**" (**NOR**) con componentes discretos, mediante la conexión en paralelo de varios **NPN** en emisor común; tales puertas dieron lugar a la primera familia lógica: **RTL** (lógica de transistores y resistencias).

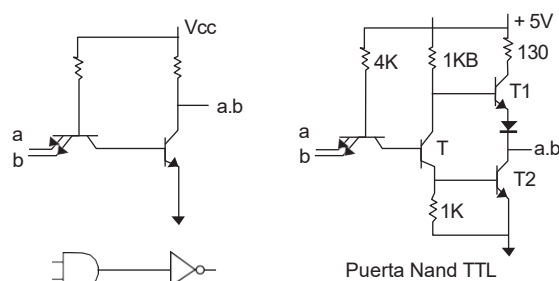


Pronto se mejoraron las características de estas puertas integradas, en cuanto a velocidad y a consumo, combinando una puerta "**Y**" de diodos con un transistor inversor en emisor común; así se configuró la puerta "**Y-NEGADA**" (**NAND**), base de la familia **DTL** (lógica de transistores y diodos) que fue la primera que llegó a alcanzar una difusión apreciable.

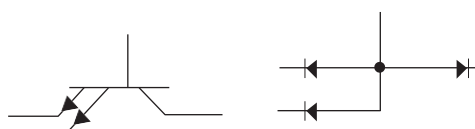


A partir de este esquema (puerta "**Y**" + inversor), aprovechando en mayor profundidad las posibilidades que ofrece la integración sobre un sustrato único, se planteó una segunda mejora en velocidad y en consumo, añadiendo una etapa de salida amplificadora de intensidad (dos transistores en push-pull) y substituyendo los diodos por un transistor multiemisor.

El resultado fue la gran familia lógica TTL (lógica de transistores con transistores).



La etapa de salida de dos transistores **NPN** (totem pole: «palo de tótem») aumenta la intensidad suministrable y disminuye la resistencia de salida; el transistor multiemisor mejora considerablemente la conmutación de la puerta (en una primera aproximación, su comportamiento puede ser analizado en términos de diodos:



La clave del funcionamiento de la puerta **TTL** es el sentido en que circula la intensidad que la base del transistor multiemisor recibe desde la resistencia de 4K:

- Si dicha corriente va «hacia fuera», es decir, si alguna de las entradas está conectada a **0**, el transistor T se encontrará en corte y el transistor T1, en colector común, transmite un **1** a la salida.

- Cuando todas las entradas se encuentran a **1** dicha intensidad circula «hacia dentro», hacia la base del transistor T, que se satura y lleva también a saturación al transistor T2, que pone la salida a **0**.

Un **0** en una entrada supone una intensidad «hacia fuera», de forma que una entrada TTL «al aire» equivale a un **1**, salvo efectos de ruido.

Las puertas **TTL** se alimentan a 5 V.

Su tensión de conmutación se sitúa en el entorno de 1,2 V, de manera que un **0** en la entrada ha de ser menor de 1 V ($V_{iLm\acute{a}x} = 1$ V) y, en , una tensión superior a 1,5 V es entendida como un **1** ($V_{iHm\acute{i}n} = 1,5$ V).

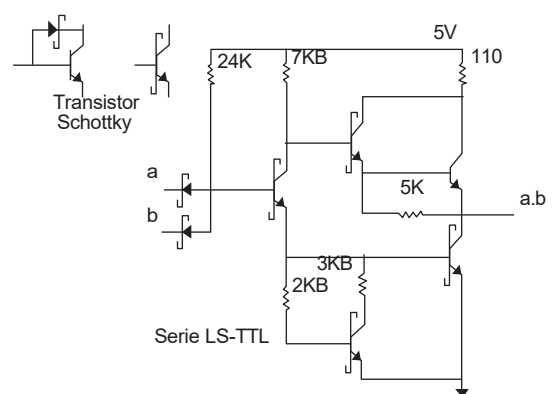
La tensión de salida para el **0** es 0 V, pero la correspondiente al **1** es de solamente 4 V. Los tiempos de propagación de la serie **TTL** estándar son del orden de 10 ns, y el consumo promedio es de unos 2 mA (10 mW).

La familia **TTL** proporcionó la base del gran desarrollo que tuvieron los sistemas digitales durante la década de los 70.

Su amplia difusión y utilización favoreció la aparición de diversas series derivadas de la mejora de características concretas, una de las cuales, la serie **LS** ha sustituido por completo a la serie estándar inicial y es la que se ha seguido utilizando a lo largo de la década de los 80.

La serie **74LS** (low power Schottky) mejora en gran medida a la serie estándar en cuanto a consumo (0,4 mA), manteniendo la velocidad de trabajo en valores análogos e incluso, algo superiores.

La disminución del consumo se deriva del empleo de resistencias de mayor valor, lo cual acarrea un aumento de las constantes de tiempo asociadas; este efecto queda compensado por la inclusión de un diodo Schottky entre base y colector de los transistores que impide su saturación (desvía la corriente de base hacia el colector antes de entrar en una saturación profunda) y, con ello, aumenta su velocidad de conmutación.



Posteriores series «avanzadas» con el mismo esquema circuital han aprovechado la reducción de dimensiones de los transistores y la correspondiente disminución de sus capacidades parásitas para conseguir tiempos de propagación inferiores: la serie **74ALS** (advanced LS) presenta tiempos por debajo de 4 ns, mientras que las series **74F** (*fast-TTL*) y **74AS** (advanced Schottky) ofrecen tiempos de propagación del orden de 2,5 ns y 1,5 ns, respectivamente, a costa de un mayor consumo (por utilizar resistencias de menor valor). Fig. 64B

“Protohistoria”



Esta línea de evolución de las puertas con transistores bipolares constituye la «edad antigua» de los circuitos integrados digitales; actualmente, apenas se utilizan las familias bipolares, salvo en determinadas aplicaciones específicas, en particular, para sistemas de muy alta velocidad.

La serie **74LS** sigue siendo útil para «recambio y» de los numerosos sistemas digitales que han sido construidos con ella (o con la serie estándar **74**), la serie **74ALS** se emplea en circuitos «interbús» y la serie **74F** resulta adecuada para diseños de muy alta velocidad de trabajo (frecuencias superiores a los 100 MHz).

CÓMO REALIZAR PRACTICAS DE ELECTRÓNICA DIGITAL

Es sencillo si disponemos de un simulador que consiste en una matriz donde se pueden insertar e interconectar circuitos integrados.

Veamos qué aspecto tiene:

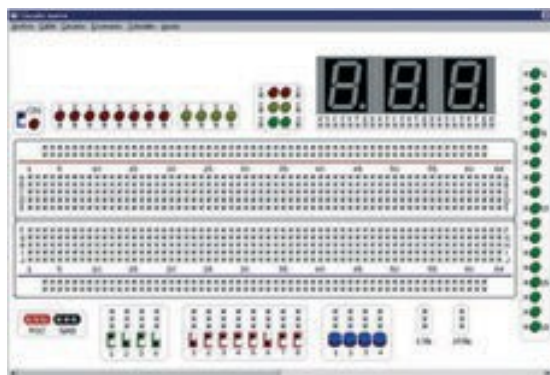


Fig. 65

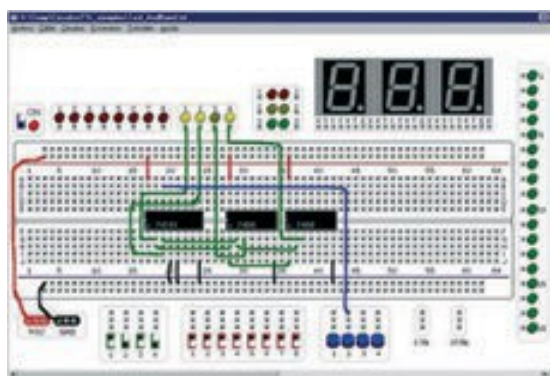


Fig. 66

ESPECULANDO CON CIRCUITOS INTEGRADOS.

Estudiemos ahora lo que ocurre cuando conectamos una función OR a una función NOT de la manera siguiente:

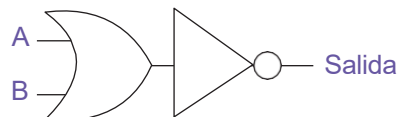


Fig. 67

¿A qué nos suena esa unión?

Aplicando todas las combinaciones posibles de unos y ceros en las entradas, obtenemos la salida para cada combinación posible tomando en cuenta las propiedades del OR y la acción inversora del NOT, con lo cual podemos construir la siguiente Tabla de Verdad:

A	B	Salida
0	0	1
0	1	0
1	0	0
1	1	0

Tenemos un circuito que produce un 1 a la salida únicamente cuando ambas entradas son 0.

Esta configuración es mejor conocida como la función NOR (la palabra NOR es una contracción de las palabras NOT-OR, que son los elementos usados para construir esta configuración) y se representa de la siguiente manera como ya hemos visto:

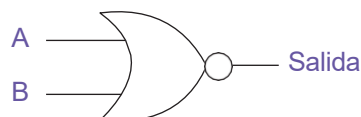


Fig. 68

A continuación, estudiemos lo que ocurre cuando conectamos una función AND a una función NOT:

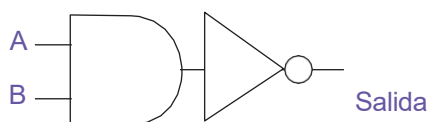


Fig. 69

La Tabla de Verdad para este circuito deberá ser como se muestra a continuación:

A	B	Salida
0	0	1
0	1	1
1	0	1
1	1	0

Tenemos un circuito que produce un 0 a la salida únicamente cuando ambas entradas son 1.

Esta configuración es mejor conocida como la función NAND (la palabra NAND es una contracción de las palabras NOT-AND, que son los elementos usados para construir esta configuración) y se representa de la siguiente manera, como ya vimos:

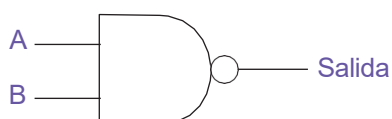
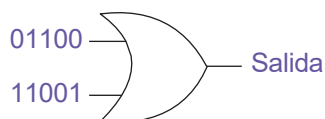


Fig. 70

Las funciones NOR y NAND son ejemplos claros que muestran cómo se pueden utilizar las tres funciones lógicas básicas para construir funciones más complejas.

Estudiemos ahora la siguiente situación:



La pregunta que nos hacemos es la siguiente: ¿Cuál será la salida del OR al introducir las palabras 01100 y 11001 en sus entradas?

Para responder a esta pregunta, notamos que los primeros bits en entrar al OR son el último bit de la palabra 01100 (esto es, un 0) y el último bit de la palabra 11001 (esto es, un 1). La salida producida por el OR será por lo tanto un 1. A continuación, los siguientes bits que entran son el penúltimo bit de la palabra 01100 (esto es, un 0) y el penúltimo bit de la palabra 11001 (esto es, un 0). La siguiente salida producida por el OR será por lo tanto 0, con lo cual a su salida ya se habrá formado la palabra 01. De esta manera, vemos que a su salida se formará la siguiente palabra:

11101

Puesto que la palabra en la salida del OR es diferente de las palabras a sus entradas, decimos que se ha llevado a cabo un procesamiento de información. Este es el propósito fundamental de todos los circuitos lógicos.

Supongamos ahora que se nos presenta un componente electrónico en el cual los niveles de voltaje naturales al sistema son +5 voltios y 0 voltios.

Dicho componente electrónico tiene dos terminales de entrada A y B y una terminal de salida. Al aplicar los siguientes niveles en sus entradas produce los siguientes niveles de voltaje en su salida:

A	B	Salida
0 volts	0 volts	0 volts
0 volts	+5 volts	+5 volts
+5 volts	0 volts	+5 volts
+5 volts	+5 volts	+5 volts

Designando al voltaje mayor (o más positivo) de +5 voltios como "1" y al voltaje menor (o más negativo) como "0", la Tabla de Verdad toma el siguiente aspecto:

A	B	Salida
0	0	0
0	1	1
1	0	1
1	1	1

El circuito se comporta como una función OR.

En este último ejemplo, muy bien podríamos haber adoptado otra convención igualmente válida.

Podríamos haber identificado al voltaje de +5 voltios con un "0" y al voltaje de cero voltios con un "1", lo que nos produce otra Tabla de Verdad diferente.

Esto es lo que se conoce como lógica negativa.

Con el propósito de evitar confusiones, nos hemos abstenido de utilizar este enfoque, aunque una vez que se hayan dominado los principios la lógica positiva y la lógica negativa son tan válidas la una como la otra.

Hagamos al menos con este ejemplo tal cosa.

Construyamos la Tabla de Verdad asignándole al nivel de +5 voltios un valor lógico de "0" y al voltaje de cero voltios un valor lógico de "1".

La Tabla de Verdad ahora será la siguiente:

A	B	Salida
1	1	1
1	0	0
0	1	0
0	0	0

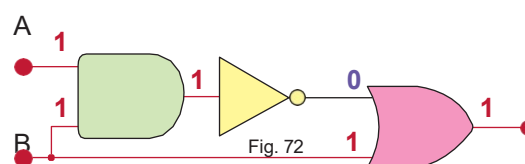
El circuito se comporta como una función AND.

Tenemos entonces que el comportamiento lógico de un circuito dependerá de las asignaciones que le demos a sus niveles de voltaje. La práctica de designar al voltaje mayor (o más positivo) como "1" y al voltaje menor (o más negativo) como "0" es conocida como **lógica positiva**.

La práctica de designar al voltaje menor (o más negativo) como "1" y al voltaje mayor (o más positivo) como "0" **es conocida como lógica negativa**.

Con las tres funciones lógicas básicas podemos construir circuitos más elaborados, de creciente complejidad, cuyo análisis se puede llevar a cabo suponiendo todas las combinaciones posibles de "unos" y "ceros" a la entrada, y siguiendo el flujo de cada combinación de valores, para ver lo que tenemos a la salida, podemos comprobar la función desempeñada por el circuito.

Por convención, los diagramas de circuitos lógicos se dibujan de modo tal que el flujo de señales es rastreado de izquierda a derecha. A continuación tenemos el diagrama de un circuito lógico simple en cuyos dos terminales de entrada A y B se han puesto dos "unos" ("1"):



En el AND del extremo izquierdo del diagrama, tenemos dos unos ("1") a la entrada, lo que genera un "1" a la salida del mismo.

Este "1" a la salida del AND es invertido por el NOT, convirtiéndose en un "0".

De este modo, tenemos a las entradas del OR en el extremo derecho del diagrama un "0" y un "1", lo que genera un "1" a la salida del mismo.

Nos falta por comprobar otras combinaciones de valores restantes, como son, $A = 0$ y $B = 0$, $A = 1$ y $B = 0$, $A = 0$ y $B = 1$.

Este método de rastreo de valores de señales se puede aplicar a cualquier circuito lógico, por complejo que sea.

Al ir construyendo circuitos lógicos cada vez más complejos, los cables que van conectados entre sí se mostrarán conectados explícitamente con un "punto" conector, mientras que los cables que simplemente se cruzan uno por encima del otro sin conectarse no tendrán el punto conector:

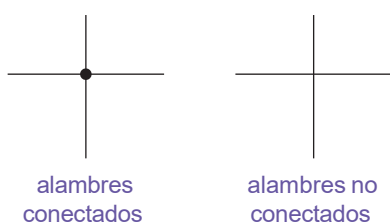


Fig. 73

Sin embargo, en las uniones tipo "T" en los diagramas esquemáticos:

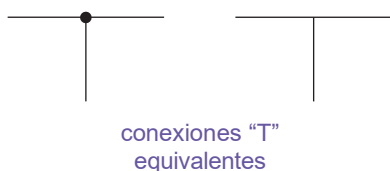
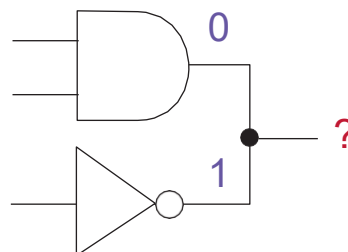


Fig. 74

Se entenderá que siempre hay una conexión entre los dos alambres, ya sea con o sin la presencia del "punto" conector (en esto hay que tener cierta precaución, ya que, en los simuladores de circuitos lógicos, en una gran variedad de programas de ordenador, es indispensable agregar siempre en los diagramas de simulación el "punto" conector, ya que muchos de estos programas no están preparados para reconocer esta convención).

En el diseño de circuitos lógicos, una de las cosas que no está permitida es conectar directamente a un mismo punto la salida de dos funciones lógicas en una forma como la que se muestra a continuación:



Puesto que se ha definido el "1" como el polo positivo de la fuente de alimentación (por ejemplo, +5 voltios) y el "0" como el polo negativo de la misma fuente (conocida vulgarmente como "tierra eléctrica"), la situación mostrada equivale ni más ni menos que a un corto circuito.

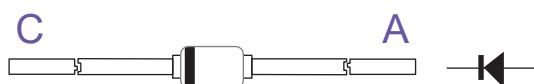
Y aunque la gran mayoría de los circuitos integrados discretos que se venden en la actualidad tienen integrada en su microelectrónica una estructura de resistencias y transistores que evitan que un corto circuito de esta naturaleza los pueda dañar, limitando el flujo de la corriente a través de los mismos, la salida de un circuito lógico de esta naturaleza es en el mejor de los casos indefinida.



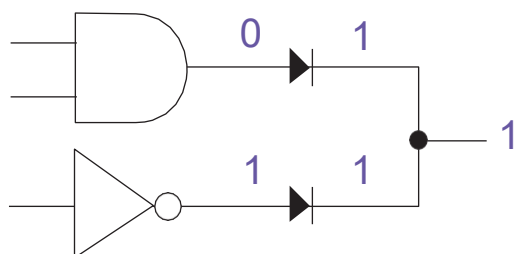
Si por alguna razón insistimos en querer conectar juntas las salidas de dos (o más) funciones lógicas básicas, lo podemos hacer utilizando diodos rectificadores, que conducen corriente eléctrica en una sola dirección más no en la dirección contraria.

Físicamente, en un circuito real, estos componentes tienen un aspecto como el que se muestra a continuación, en donde la banda puesta en un extremo del componente indica la polaridad del cátodo (la parte del componente

que va conectada al polo negativo de la fuente de alimentación cuando se desea que este componente conduzca corriente eléctrica), correspondiendo con banda tocada por la flecha en el símbolo representativo del diodo que aparece a la derecha:



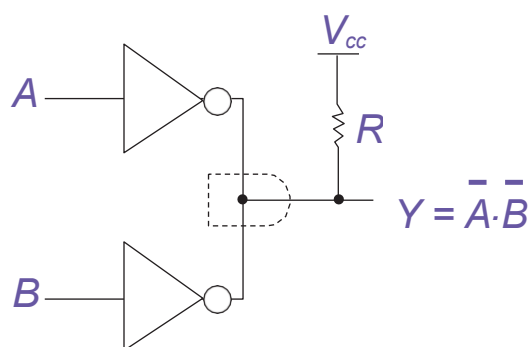
Usando diodos rectificadores, el circuito lógico anterior ya "corregido" presentaría el siguiente aspecto:



En este circuito el AND superior está poniendo un "0" en su terminal de salida mientras que el NOT inferior está poniendo un "1" en la suya propia. El voltaje relacionado con el "1" lógico normalmente produce una corriente eléctrica que fluye hacia el "0" lógico. Pero en los diodos rectificadores, la corriente eléctrica solo puede fluir en la dirección que marca la flecha, de modo tal que si en la cola de la flecha hay un "0" y en la punta de la flecha hay un "1" como ocurre con el AND superior, el diodo no permitirá el paso de la corriente y actuará como si el alambre estuviera cortado, efectivamente desconectando al AND de la salida combinada. En otras palabras, el "1" del inversor NOT es el que gana. Pero si la salida del NOT fuese "0" y la salida del AND fuera "1", entonces el AND ganaría. Y si la salida de ambos fuera "1", entonces ambos ponen

un "1" lógico en la línea de salida y no hay contradicción lógica alguna (ni corto circuito). Pero si repasamos estas funciones, nos daremos cuenta de que la inserción de los diodos en el circuito hace que entre ambos proporcionen el equivalente de la función OR, con la desventaja de que los diodos no proporcionan ninguna amplificación de la señal. Las buenas prácticas de diseño nos indican remover los dos diodos y reemplazarlos por un bloque OR.

Existe una excepción importante a la prohibición de conectar directamente a un mismo punto las salidas de dos o más bloques lógicos, y es cuando se están utilizando circuitos integrados capaces de realizar algo que se conoce como lógica cableada (wired logic), también conocida como la lógica de colector abierto (open collector logic), que tiene que ver con el hecho de que tales circuitos, contruidos a base de transistores bipolares (ya sabemos que un transistor bipolar básico tiene tres terminales, la base, el emisor y el colector), utilizan el terminal denominado colector del transistor bipolar en modo abierto (sin conexión previa) para llevar a cabo esta función, estando por ello diseñados para conectarse a un mismo punto al que además hay que agregar una resistencia conectada al polo positivo de la fuente de alimentación, como se muestra en el siguiente esquema en el que se han conectado las salidas de dos bloques NOT:



Por regla general, al utilizarse la lógica cableada ó "lógica de colector abierto", en el punto de unión se lleva a cabo la función AND. Esta es la razón por la cual en el dibujo

anterior se ha dibujado en trazos una AND alrededor del punto de unión de las salidas de los NOTs. Si tenemos un "0" a la entrada de ambos NOTs, la salida de los dos será "1", y por la acción de la AND, en el punto de unión, la salida Y del circuito tendrá un valor "1". Si cualquiera de las entradas A ó B a los NOTs es "1", será invertido a "0", y por la acción AND sobre la salida de ambos la salida común Y tendrá un valor "0". Si ambas entradas son "1", la salida Y será "0".

Estamos ante un circuito en el cual la salida será "1" únicamente cuando ambas entradas son "0", lo que es en efecto la acción de un circuito NOR. No todas las familias de circuitos lógicos integrados pueden conectarse de esta manera. Obsérvese que dentro de los bloques NOTs se dibujó el símbolo de un pequeño diamante con una barra horizontal puesta debajo del mismo. Este es el símbolo utilizado para indicar en los diagramas esquemáticos que se están usando componentes trabajando bajo lógica cableada.

Está así especificado por la convención técnica IEEE/ANSI-1984 que indica que cualquier componente, cuya salida pueda ser cableada a otro componente, deberá destacarse dibujando dentro del bloque lógico cerca de la salida del mismo el símbolo del diamante con la barra horizontal puesta debajo:

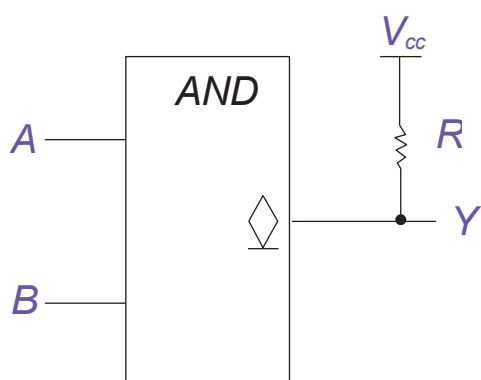


Fig. 79

Hemos visto que para poder aprender se utilizan los simuladores que permiten la posibilidad de realizar muchas maniobras en plan experimental antes de construir la definitiva.

Gracias a la potencia de los ordenadores existe una alternativa, mucho más económica y mucho más rápida para construir un prototipo utilizando varias combinaciones de circuitos lógicos, consistente en usar un simulador de "software", o sea un programa de ordenador que permite seleccionar los símbolos lógicos convencionales de un catálogo para irlos conectando en la pantalla. Conseguimos llevar a efecto la simulación de un experimento virtual en donde no hay un solo cable a la vista ya que todo se lleva a cabo esquemáticamente desde el teclado.

Uno de dichos simuladores es el Digital Works, que tiene un costo aproximado de ochenta dólares:

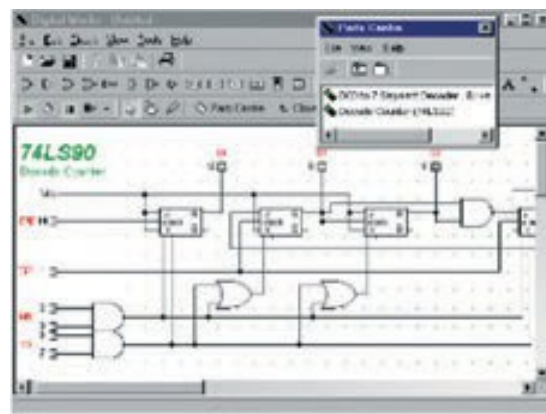
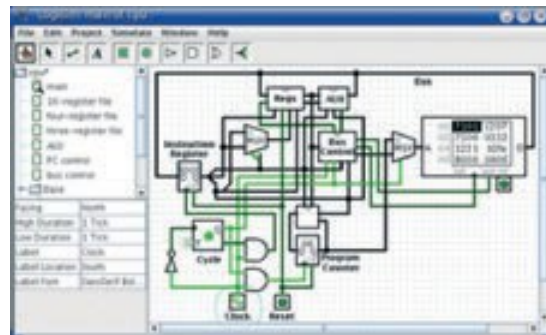
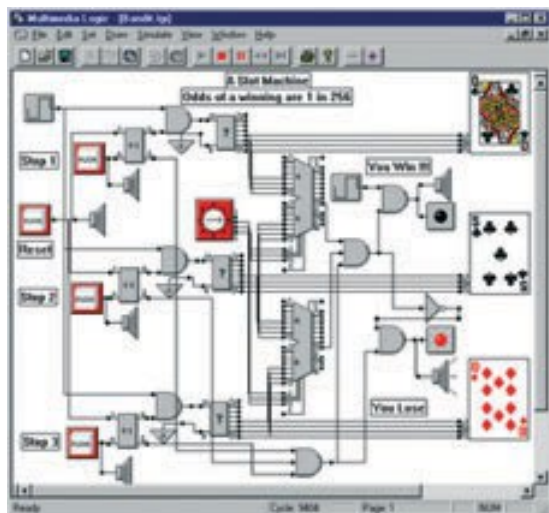


Fig. 80

Existen también simuladores de alta potencia que pueden ser descargados gratuitamente de Internet. Uno de ellos es Logisim, desarrollado en 2002 por Carl Burch, que puede trabajar en cualquier ordenador que tenga instalada la plataforma Java (hoy la gran mayoría la tiene, y los que no la tienen la pueden obtener también gratuitamente de Internet):



Hay otro programa que se puede descargar gratuitamente. Es el programa Multimedia Logic, y que presenta el siguiente aspecto:



Otra opción extremadamente buena para construir circuitos lógicos (simulados) en la pantalla de un ordenador es la versión estudiantil del programa PSpice.

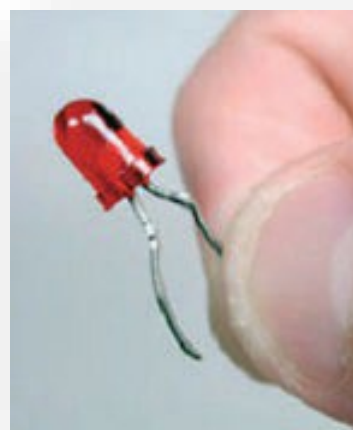
Se trata de un programa tan completo y sofisticado que el archivo de descarga tiene un tamaño de 28 Megabytes, de modo que la descarga del mismo no es recomendable a través de una conexión Internet de baja velocidad.

Además de los programas anteriormente mencionados que nos permiten diseñar y construir circuitos lógicos (simulados), existe otra alternativa inmediata para poder ver la acción de los circuitos lógicos "en vivo" de forma interactiva, como si se tuviese a mano un tablero con los componentes físicos reales. A modo de ejemplo, se recomienda visitar el siguiente sitio en el cual el usuario puede jugar a través del ratón del ordenador con varias compuertas lógicas, encendiendo con unos o apagando con ceros las terminales de entrada, viendo de inmediato los efectos a la salida (o salidas) de los circuitos lógicos en diodos emisores de luz (LED) simulados:

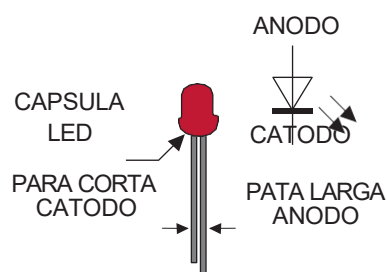
<http://www.dstcorp.com/James/LogicPrimer/Gates.html>

Los circuitos lógicos interactivos que aparecen en este último enlace fueron creados precisamente con el programa MultiMedia Logic ya mencionado.

Aunque no desempeña ninguna función lógica, un componente que aparece con cierta frecuencia en el mundo de la lógica digital como indicador luminoso para proporcionar algún tipo de información visual al usuario es el diodo emisor de luz (light emitting diode ó LED), que ya estudiamos en su día pero que interesa de nuevo recordar por su útil adaptación a los circuitos lógicos.



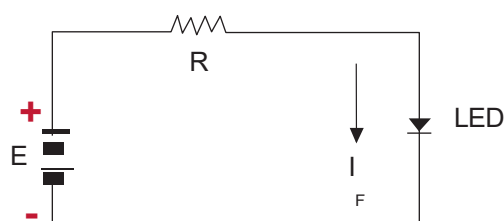
No sólo es un diodo en el sentido estricto de la palabra (un componente en el cual la corriente eléctrica puede fluir únicamente en un sentido, pero no en el sentido contrario) sino que además es capaz de emitir luz de cierto color cuando se le aplica un voltaje con la polaridad correcta (con la polaridad del voltaje invertida, el LED no conducirá corriente alguna ni emitirá luz). El diodo LED generalmente se construye con un terminal más corto (denominado cátodo ó terminal -) que el otro (denominado ánodo ó terminal +). A continuación figura un LED junto con el símbolo con el que se le representa en los diagramas esquemáticos:



El diodo emisor de luz LED está disponible en varios colores, ya sea rojo, verde, azul, ámbar o blanco.

El esquema más sencillo para encender un diodo emisor de luz LED con una batería común y corriente E es el siguiente, en el que basta aplicarle el voltaje de la batería poniendo atención a la polaridad apropiada:

Circuito básico para encender un diodo LED

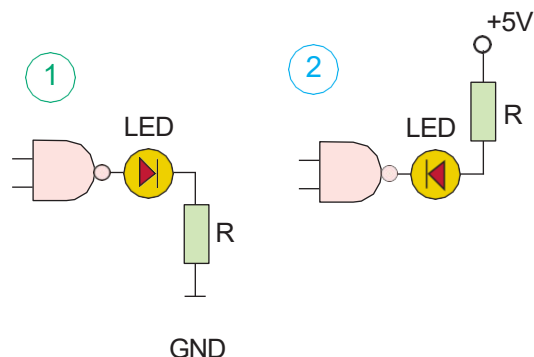


$$I_F = \frac{E - V_F}{R}$$

Fig. 84

La resistencia R , mostrada en el esquema, tiene como objetivo limitar la magnitud de la corriente eléctrica impidiendo que haya una sobreintensidad que dañaría al componente y al sistema, pero también tiene como finalidad fijar el valor de la corriente con la cual el LED podrá emitir su mayor brillo posible sin dañarse. Por no ser el diodo LED un diodo ideal sino un diodo real (el diodo ideal es aquél que al estar polarizado en un sentido actúa como un circuito abierto y al estar polarizado en el sentido inverso actúa como un circuito cerrado sin pérdidas de voltaje a través del mismo), cuando está conduciendo existe una pequeña caída de voltaje constante V_F (forward voltage) a través del mismo, que tiene que ser restada del voltaje de la batería E para determinar la magnitud de la corriente eléctrica que fluirá a través del mismo según la fórmula.

Una gran ventaja de los diodos emisores de luz LED es que, por la pequeña cantidad de corriente eléctrica que requieren para lucir, se pueden conectar directamente a la salida de un circuito lógico ya sea para indicar una salida de "0" ó una salida de "1". Hay dos formas en las cuales se puede llevar a cabo la conexión, indicadas en los siguientes dibujos:



En el diagrama (1) de la izquierda, cuando la salida de la NAND es "1" en realidad este componente estará poniendo un voltaje de algo así como unos +5 V directamente a la entrada del diodo LED, y como el otro extremo del LED está conectado a través de la resistencia R al nivel de "0" ó tierra eléctrica (GND), estará polarizado justo en la forma correcta para encenderse indicando la presencia del "1" lógico a la salida de la NAND. La resistencia R es utilizada para impedir un cortocircuito y limitar la magnitud del corriente justo a lo que requiere el diodo LED para poder encenderse adecuadamente. Si la salida de la NAND es "0" en realidad este componente no estará poniendo ningún voltaje a su salida que será de cero voltios, por lo que ambos extremos del LED estarán conectados a tierra y el LED estará apagado, indicando la presencia de un "0" lógico. En lo que respecta al diagrama (2) de la derecha, cuando la salida de la NAND es "1" estará poniendo un voltaje de +5 V en un terminal del LED, y como el otro terminal del LED también está conectada al mismo voltaje de +5 V el LED no conducirá corriente eléctrica alguna, permaneciendo apagado, ya que para poder encenderse el diodo LED tiene que tener su ánodo conectado a un "1" lógico y su cátodo conectado a un "0" lógico para que la corriente eléctrica pueda fluir en la dirección indicada por la flecha de su símbolo, no luciendo con ambos terminales conectados a un "1" lógico.

Cuando la salida de la NAND es un "0" lógico, se polariza al LED en la forma correcta y la corriente eléctrica puede fluir de "1" a "0" entrando por el terminal de salida de la NAND. (No hay ninguna contradicción en que el terminal de salida de la NAND actúe al mismo tiempo como una entrada de corriente

eléctrica, ya que una cosa es el voltaje de cero voltios puesto a la salida y otra cosa es la corriente eléctrica que le puede entrar a través de su terminal de salida; recuérdese siempre que los "ceros" y "unos" están definidos en función de la presencia o la ausencia de un voltaje.) La principal diferencia entre el circuito (1) de la izquierda y el circuito (2) de la derecha es que en el circuito de la izquierda el LED se enciende cuando la salida de la NAND es "1" y se apaga cuando la salida del LED es "0", mientras que en el circuito de la derecha el LED se apaga cuando la salida de la NAND es "1" y se enciende cuando la salida es "0".

Posiblemente una de las aplicaciones más útiles que pueda tener un diodo emisor de luz LED para el técnico especializado en dar mantenimiento a sistemas digitales sea en la construcción de la herramienta fundamental utilizada por dichos técnicos: la punta de prueba lógica (logic probe), como la siguiente:



Este tipo de puntas de prueba generalmente tienen dos diodos emisores de luz LED, uno para indicar una condición lógica de "0" (LOW ó LO) y el otro para indicar una condición lógica de "1" (HIGH ó HI). Generalmente no requieren de fuente de energía interna (pilas), ya que toman su energía directamente del mismo circuito que está siendo analizado a través de unos terminales de tipo caimán. En la punta de prueba mostrada arriba, la pinza caimán de color rojo se conecta al polo positivo (+) de la fuente de alimentación, mientras que la pinza de color negro se conecta al polo negativo (-) de la fuente. El interruptor selector en esta punta de prueba, identificado como "TTL/CMOS", sirve para escoger los niveles adecuados de voltaje y corriente según sea la familia lógica a la cual pertenezca el componente que está siendo analizado, ya sea TTL ó CMOS. Aunque en manos de un técnico experto la punta de prueba lógica arriba mostrada sirve admirablemente para el análisis, diagnóstico y reparación de la gran mayoría de los sistemas digitales en uso comercial hoy en día, para ciertas

aplicaciones sumamente especializadas, donde el presupuesto no es ningún problema, existen otras puntas de prueba mucho más refinadas que en realidad son puntas de prueba lógicas sólo de nombre, ya que por su complejidad se trata en realidad de instrumentos de medición de alta precisión, como la "punta de prueba lógica" que aparece en la fotografía:

